(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－320191

(43)公開日　平成10年(1998)12月４日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/06	５５０	Ｇ０６Ｆ　9/06	５５０Ａ
Ｇ０９Ｃ　1/00	６３０	Ｇ０９Ｃ　1/00	６３０Ｚ
	６６０		６６０Ｄ

審査請求　未請求　請求項の数5　ＦＤ　（全 21 頁）

(21)出願番号　特願平10－132755

(22)出願日　平成10年(1998)４月27日

(31)優先権主張番号　０８／８４１，３１４

(32)優先日　1997年４月30日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009597
モトローラ・インコーポレイテッド
MOTOROLA INCORPORATED
アメリカ合衆国イリノイ州シャンバーグ、イースト・アルゴンクイン・ロード1303

(72)発明者　ディビッド・マイケル・ハリソン
アメリカ合衆国アリゾナ州85203、メサ、イースト・ローレル・ストリート　1825

(74)代理人　弁理士　池内　義明

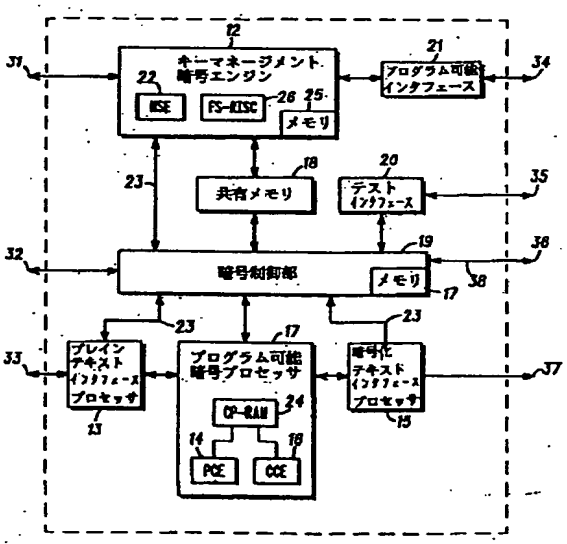
最終頁に続く

(54)【発明の名称】　プログラム可能暗号処理システムおよび方法

(57)【要約】

【課題】　単一のＵＬＳＩダイ上で実施できるいくつかの処理資源１４，１６，２６を含む改善されたプログラム可能暗号処理システム１０を実現する。

【解決手段】　本処理システムはキーおよびアルゴリズムの双方に対して機敏であり現プログラムの実行の間の次のプログラムのバックグランドステージングおよびコンテクスト（キーおよび状態）により種々の暗号プログラムの同時的実行を可能にする。本プログラム可能暗号処理システムはチャネルプログラムにしたがったデータユニットの処理ためのプログラム可能暗号プロセッサ１７、チャネルプログラムを識別するための暗号制御部１１、外部ホストに対し非同期的にデータユニットを転送しかつ受信するための２つのインタフェースプロセッサ１３，１５を含む。データユニットは特定のチャネルプログラムを識別し、かつ識別されたチャネルプログラムにしたがって選択された処理エンジンで処理される。



【特許請求の範囲】

【請求項１】　プロセッサ可能暗号処理システム（１０）であって、

データユニットを処理するためのプログラム可能暗号プロセッサ（ＰＣＰ）（１７）、そして各々のデータユニットに含まれる情報に基づき各々のデータユニットに対するチャネルプログラムを識別するための暗号制御装置（ＣＣ）（１１）、

を具備することを特徴とするプログラム可能暗号処理システム（１０）。

【請求項２】　前記データユニットの各々はヘッダフィールド、コマンドフィールドおよびペイロード部分を含み、前記ＣＣは、前記データユニットの１つのヘッダフィールドを読み取るための手段、

前記ヘッダフィールドにおけるチャネルインデクスに基づき前記１つのデータユニットを処理するために複数のチャネルプログラムから前記チャネルプログラムを識別するための手段、

前記チャネルインデクスに応じて、前記チャネルプログラムが前記ＰＣＰにおける処理エンジンにダウンロードされるようにするための手段、

前記処理エンジンによる処理を予期して前記ペイロード部分を前記ＰＣＰに転送するための手段、

を具備し、前記コマンドフィールドは前記処理エンジンによって前記１つのデータユニットに対して行なわれるべき機能を識別し、かつ前記ＰＣＰはさらに、

前記１つのチャネルプログラムを記憶するための第１のメモリ、

前記処理エンジンによる前記ペイロード部分の処理に先立ち前記ペイロード部分を記憶するための第２のメモリ、

前記機能を決定するために前記１つのデータユニットの前記コマンドフィールドを読み取るための手段、そして前記チャネルプログラムを前記機能の実行のために前記処理エンジンにロードするための手段、

を具備し、前記データユニットはヘッダフィールド、コマンドフィールドおよびペイロード部分を含み、かつ前記ＰＣＰは前記ペイロード部分を記憶するための第１のメモリおよび複数のチャネルプログラムを記憶するための第２のメモリを含み、

前記チャネルプログラムの１つは前記データユニットの前の処理の間に前記データユニットの１つの処理を予期して処理エンジンの前記第２のメモリにダウンロードされることを特徴とするプログラム可能暗号処理システム。

【請求項３】　ヘッダ部分、コマンド部分および関連するペイロード部分を有するデータユニットを処理するためのデータユニット処理システムであって、前記ヘッダ部分は前記関連するデータユニットを処理するためのチャネルプログラムを識別し、かつ前記コマンド部分は前記関連するデータユニットのペイロード部分に対して実行されるべき機能を識別し、前記システムは、

前記データユニットの各々によって特定される前記チャネルプログラムにしたがって前記データユニットの各々を処理するためのプログラム可能暗号プロセッサ（ＰＣＰ）（１７）、そして前記ヘッダ部分を読み取りかつ前記関連するデータユニットによって識別される前記チャネルプログラムを前記ＰＣＰにおける処理エンジンにダウンロードさせる暗号制御部（ＣＣ）（１１）であって、該ＣＣは前記ペイロード部分が前記チャネルプログラムによる処理を待機するため前記処理エンジンのメモリに転送されるようにするもの、

を具備することを特徴とするデータユニット処理システム。

【請求項４】　複数の処理エンジンを有する処理システムにおいてデータユニットを処理する方法（２００）であって、

（２０８）前記データユニットの第１のものにおける情報に基づき複数のチャネルプログラムからあるチャネルプログラムを識別する段階、

（２０８）前記第１のデータユニットを処理するために前記複数の処理エンジンからある処理エンジンを識別する段階、

（２１０）前記識別された処理エンジンに関連するメモリに前記第１のデータユニットを導く段階、

（２１６）前記識別された処理エンジンに前記識別されたチャネルプログラムをロードする段階、そして

（２２０）前記識別されたチャネルプログラムを使用して前記識別された処理エンジンにおいて前記第１のデータユニットを処理する段階、

を具備し、前記チャネルを識別する段階はさらに前記チャネルに関連するコンテクストを識別する段階を含み、該コンテクストはメモリに記憶され、かつ前記処理する段階は前記識別されたチャネルプログラムによって前記第１のデータユニットを処理する段階を含み、前記識別されたチャネルプログラムは前記関連するコンテクストを使用することを特徴とする複数の処理エンジンを有する処理システムにおいてデータユニットを処理する方法（２００）。

【請求項５】　複数の処理ユニットを有するプログラム可能暗号処理システムにおける暗号機能を同時的に行なう方法（２００）であって、

（２０２）第１のヘッダフィールド、コマンドＩＤフィールドおよびペイロード部分を含む第１のデータユニットを受信する段階、

（２０８）前記第１のヘッダフィールドに基づき前記第１のデータユニットに対して前記暗号機能の１つを実行するために前記処理ユニットの１つを選択する段階、

（２１０）前記第１のデータユニットを前記選択された１つの処理ユニットに導く段階、

前記選択された１つの処理ユニットが前記コマンドＩＤフィールドにおける情報に基づき前記ペイロード部分に対し前記暗号機能の選択された１つを実行する（２２０）段階、そして前記実行する段階の達成の間にインタフェースプロセッサにおいて第１の処理されたデータユニットを形成する段階、

を具備し、かつ前記方法は前記第１の処理されたデータユニットが形成されたことを外部ホストに通知する段階を含み、そして前記導く段階は前記第１のデータユニットを前記選択された１つの処理ユニットに関連するメモリに導く段階を含むことを特徴とする複数の処理ユニットを有するプログラム可能暗号処理システムにおける暗号機能を同時的に実行する方法（２００）。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は一般的には保安暗号通信の分野に関する。

【０００２】

【従来の技術】通信市場における傾向は明らかに商業用および軍事用マーケットの双方に対して保安性（ｓｅｃｕｒｉｔｙ）の必要性を規定している。通信システムが複雑な通信サービスおよび能力を備えてより精巧になるに応じて、情報を安全にまたは保安されて保つことが重要である。保安機器に伴う問題の１つはリバースエンジニアリング技術による搾取からの暗号プログラムの保護である。暗号プログラムがハードウエアに組み込まれている暗号プログラムのハードウエアによる実施は一般に安全であると考えられる。ハードウエアの実施に伴う問題は敵対者または利害の対立するものが非常な努力を用いてダイ探査（ｄｉｅ　ｐｒｏｂｉｎｇ）および分析によりプログラムを決定できることである。ハードウエアで実施されるまたは構成される暗号システムの他の問題は暗号プログラムを処理するチップのための高いコストの半導体処理である。半導体は保安状態の下で製造され、それは暗号プログラムがハードウエア論理に組み込まれているためである。

【０００３】ソフトウエアによって実施されるまたは構成される暗号プログラムは、しかしながら、典型的にはハードウエアの構成ほど安全でないと考えられ、それはソフトウエアのアクセス可能性のためである。ソフトウエアの構成に伴う典型的な問題は複数プログラムの同時処理が結果として保安オペレーティングシステムにおけるタスク交換（ｔａｓｋ　ｓｗａｐｐｉｎｇ）による性能の損失を生じることである。ソフトウエアの構成に伴う他の問題は典型的なマイクロプロセッサおよびデジタル信号プロセッサの算術論理ユニットが高速暗号処理にとって望ましい高速の並列、数値および論理処理資源を持たないことである。

【０００４】

【発明が解決しようとする課題】ハードウエアおよびソフトウエア双方の暗号処理システムに伴う問題はサブシステムの間で交換されたときキー変数データの無防備さまたは攻撃されやすさ（ｖｕｌｎｅｒａｂｉｌｉｔｙ）である。これは今日の暗号システムにとって一般的な保安性の危険である。

【０００５】従って、必要なことは改善された暗号処理システムおよび方法である。さらに必要なことは暗号プログラム（ｃｒｙｐｔｏ　ｐｒｏｇｒａｍｓ）を含まずかつ商業的な半導体工場で処理されて半導体処理のコストを低減できる暗号処理システムおよび方法である。また、高性能の暗号プログラム処理のための暗号システムも必要である。さらに、同時に複数のプログラムを実行できる暗号システムも必要である。さらに必要なことは、キーおよびアルゴリズムに機敏な（ｋｅｙａｎｄ　ａｌｇｏｒｉｔｈｍ　ａｇｉｌｅ）暗号処理システムおよび方法である。さらに必要なことは、迅速かつ安全に処理される各々のデータユニットに対するコンテクスト（ｃｏｎｔｅｘｔ）およびプログラム（例えば、アルゴリズム）を切り替えできる暗号処理システムおよび方法である。さらに、異なるサブシステムの間で交換されたときキー変数データを保護する暗号システムテが必要である。さらに必要なことは、リバースエンジニアリングから暗号プログラムが保護される暗号システムである。

【０００６】

【課題を解決するための手段】本発明は、とりわけ、プログラム可能な暗号処理システムおよび方法を提供する。本発明はまた高性能の暗号プログラムを処理するのに適した暗号処理システムを提供する。本発明はまた同時に複数の暗号プログラムを処理するシステムおよび方法を提供する。本発明はまた処理される各々のデータユニットに対するコンテクストおよびプログラム（例えば、アルゴリズム）を高速かつ安全に切り替える暗号処理システムおよび方法を提供する。本発明はさらに、異なるサブシステムの間で交換されたときキー変数データを保護する暗号処理システムおよび方法を提供する。本発明はまたフェイルセイフのアーキテクチャにおける暗号プログラムを処理するのに適したシステムおよび方法を提供する。本発明はさらに典型的な暗号処理システムに関連する半導体処理コストを低減するプログラム可能な暗号処理システムを提供する。好ましい実施形態では、キー変数データの保安性がサブシステムの間で交換される場合に保護される。また、好ましい実施形態においては、暗号プログラムは現場に配備された機器において更新できる。また、好ましい実施形態では、暗号プログラムはリバースエンジニアリングから保護される。

【０００７】

【発明の実施の形態】本発明は特に添付の特許請求の範囲に指摘されている。しかしながら、本発明のより完全な理解は添付の図面と共に以下の詳細な説明および特許請求の範囲を参照することにより得ることができる。図

面においては同じ参照数字は図面にわたり同様の項目に言及している。

【０００８】図１は、本発明の好ましい実施形態に係わるプログラム可能な暗号処理システム（ｃｒｙｐｔｏ ｐｒｏｃｅｓｓｉｎｇ ｓｙｓｔｅｍ）のハードウエアブロック図を示す。暗号処理システム１０は、好ましい実施形態では、２つの主な処理要素、キーマネジメント暗号エンジン（Ｋｅｙ ｍａｎａｇｅｍｅｎｔ ｃｒｙｐｔｏ ｅｎｇｉｎｅ：ＫＭＣＥ）１２およびプログラム可能暗号プロセッサ（ｐｒｏｇｒａｍｍａｂｌｅ ｃｒｙｐｔｏｇｒａｐｈｉｃ ｐｒｏｃｅｓｓｏｒ：ＰＣＰ）１７を有する。ＰＣＰ１７は２つの処理エンジン、プログラム可能暗号エンジン（ｐｒｏｇｒａｍｍａｂｌｅ ｃｒｙｐｔｏｇｒａｐｈｉｃ ｅｎｇｉｎｅ：ＰＣＥ）１４および構成可能暗号エンジン（ｃｏｎｆｉｇｕｒａｂｌｅ ｃｒｙｐｔｏｇｒａｐｈｉｃ ｅｎｇｉｎｅ：ＣＣＥ）１６を具備する。前記処理エンジンはチャネルプログラムの実行を行う。システム１０はまた暗号コントローラ（ｃｒｙｐｔｏｇｒａｐｈｉｃ ｃｏｎｔｒｏｌｌｅｒ：ＣＣ）１１を含み、該ＣＣ１１は処理エンジンのためのプログラム管理を行う。システム１０はまた外部インタフェースおよびシステム１０のためのシグナリングを提供するプレインテキストまたは平文インタフェースプロセッサ（ｐｌａｎｅ ｔｅｘｔ ｉｎｔｅｒｆａｃｅ ｐｒｏｃｅｓｓｏｒ：ＰＴＩＰ）１３および暗号文インタフェースプロセッサ（ｃｉｐｈｅｒ ｔｅｘｔ ｉｎｔｅｒｆａｃｅ ｐｒｏｃｅｓｓｏｒ：ＣＴＩＰ）１５を含む。前記インタフェースプロセッサはまた外部ホストとシステム１０の内部処理システムの間の高性能保安フレキシブルバッファを提供する。システム１０はまたＫＭＣＥ１２およびＰＣＰ１７の間の弾力性あるバッファとして作用する共用または共有メモリ１８を含む。システム１０はまたＦＩＬＬおよびＣＩＫポート３４に結合されたプログラム可能インタフェース２１を含む。システム１０の試験はオンチップエミュレーションおよびＪＴＡＧポート３５を含む試験インタフェース２０を使用して行うことができる。

【０００９】ＫＭＣＥ１２は内部メモリ２５を含みかつ内部バス２３によってＣＣ１１に結合されている。他の内部バス２３はＰＴＩＰ１３、ＣＴＩＰ１５、ＰＣＰ１７および共有メモリ１８をＣＣ１１に結合する。

【００１０】好ましい実施形態では、ＫＭＣＥ１２はまたフェイルセイフの縮小命令セットコンピュータ（ＦＳ－ＲＩＳＣ）２６を含む。ＫＭＣＥ１２は好ましくはモッドまたはモジュロＮ解抽出器（ｍｏｄ Ｎ ｓｏｌｕｔｉｏｎ ｅｘｔｒａｃｔｏｒ：ＮＳＥ）２２のような第２の処理資源を含む。ＦＳ－ＲＩＳＣ２６は好ましくは２重の（ｄｕａｌ）３２ビットＲＩＳＣコアからなり、これは組み込まれたまたは埋込まれた（ｅｍｂｅｄｅｄ）保安オペレーティングシステム（ｓｅｃｕｒｅ ｏｐｅｒａｔｉｎｇ ｓｙｓｔｅｍ：ＳＯＳ）を実行する。該保安オペレーティングシステムはタスクがシステム１０の外部のプログラムメモリから実行できるようにするためセグメンテイション（ｓｅｇｍｅｎｔａｔｉｏｎ）およびタスク管理を提供する。そのようなタスクは保安処理を行わないまたは微妙なデータ（ｓｅｎｓｉｔｉｖｅ ｄａｔａ）を取り扱わないタスクおよびサブルーチンを含むことができる。保安処理を行いあるいは微妙なデータを取り扱うタスクおよびサブルーチンは好ましくはメモリ２５に含まれる内部プログラムメモリ（ＲＯＭ）から実行される。

【００１１】本発明の好ましい実施形態では、前記ＦＳ－ＲＩＳＣのＳＯＳによって内部ＲＯＭから行われる機能は、とりわけ、システム１０のマスタコントロール、システム１０のセルフテストおよび警報監視、プログラムロードおよび実時間マルチレベル保安タスク管理を含む。プログラムロードは保安および非保安プログラムの双方を内部メモリ２５へロードするかあるいはアルゴリズムまたはプログラムのＰＣＰ１７へのロードを含む。

【００１２】ＦＳ－ＲＩＳＣ２６はまたメモリ２５の内部プログラムメモリ（ＲＡＭ）からのアプリケーションソフトウエアを動作させることができる。内部プログラムＲＡＭからＦＳ－ＲＩＳＣ２６によって動作する典型的なアプリケーションソフトウエアはＣＩＫおよび微妙なデータの低レベル処理のためのフィルポート処理またはポート充填処理（ｆｉｌｌ－ｐｏｒｔ ｐｒｏｃｅｓｓｉｎｇ）のような機能を含む。この例はキーのロードを含む。動作する他のアプリケーションソフトウエアの例は、例えば、パブリックキープログラムによるセッションキーの発生および他のキー管理および制御機能を含む。アプリケーションソフトウエアはまたロード（ｌｏａｄｉｎｇ）、検証（ｖｅｒｉｆｙｉｎｇ）、変更（ｃｈａｎｇｉｎｇ）および会計検査（ａｕｄｉｔｉｎｇ）のようなシステム管理およびキー管理機能を含むことができる。

【００１３】ＦＳ－ＲＩＳＣ２６はまた外部プログラムメモリからのアプリケーションソフトウエアを動作させることができる。これらの外部プログラムメモリは外部ホストシステムのＲＡＭとすることができる。外部プログラムＲＡＭから動作するそのようなアプリケーションソフトウエアは好ましくはインタフェースプロトコル処理（たとえば、ＤＳ－１０１およびＮＳＡ８７－２７）、キー管理オペレーション、コマンド処理、非保安プログラムソフトウエアおよび微妙なデータの処理に直接関連しないソフトウエアのような機能を含む。

【００１４】ＰＣＰ１７は、とりわけ、データユニットに関する機能を行いかつデータユニットを処理する高性能プログラム可能スーパースケーラ（ｓｕｐｅｒｓｃａｌｅｒ）暗号処理要素である。データユニットは、好ましくは外部ホストにより、インタフェースプロセッサ１

３へとまたはインタフェースプロセッサ１５へとロードされる。ＣＣ１１は要求されるコンテクスト（ｃｏｎｔｅｘｔ）、プログラムコード、状態（ｓｔａｔｅ）および変数をデータユニットのヘッダ情報の読取りに応じてインスタンシエイト（ロード）することによりデータユニットの処理を開始する。いったんデータユニットがＰＣＰ１７にロードされかつ処理が行われると結果が出力インタフェースプロセッサに書き込まれる。処理されたデータユニットはあるいはさらに処理を行うためにＫＭＣＥ１２のような他のデスティネイションに提供することができる。

【００１５】ＣＣ１１は、とりわけ、インタフェースプロセッサ１３および１５および暗号エンジン１４および１６、ＮＳＥ２２およびＦＳ－ＲＩＳＣ２６の実行資源の間で総合的なデータ移動を管理する。ＣＣ１１は概略的に移動すべきデータ、ＰＣＰ１７にインストールすべきタスク、およびいつプログラムの実行を開始するかを決定することによって安全なまたは保安実時間オペレーティングシステムとして動作する。ＣＣ１１はこれを各々のデータユニットの内容を調べることにより達成する。これは後に詳細に説明する。このデータ駆動アーキテクチャはシステム１０に高性能の処理能力を提供する。さらに、ＣＣ１１はバックグランド準備作業またはバックグラウンドステージング（ｂａｃｋｇｒｏｕｎｄ　ｓｔａｇｉｎｇ）を行う。次のタスクおよびデータユニットは現在のタスクの実行の間に設定または準備される（ｓｔａｇｅｄ）。前記バックグランド作業はシステム１０のための高いスループットを可能にする。例えば、ＰＣＰ１７へのデータユニットの転送、メモリのクリーンアップ、および次のデータユニットのためのプログラムロードが前のデータユニットの処理の間に行われる。

【００１６】本発明の好ましい実施形態では、ＰＣＰ１７は、とりわけ、チャネル暗号化および暗号解読のような機能および保安通信およびシグナリングにおいて典型的に行われる他のデータ処理を行う、２つの高速処理エンジン、ＰＣＥ１４およびＣＣＥ１６を具備する。好ましい実施形態では、ＰＣＥ１４はコードブック形式の（ｃｏｄｅｂｏｏｋ　ｓｔｙｌｅ）プログラムを行い、一方ＣＣＥ１６はコンバイナ形式の（ｃｏｍｂｉｎｅｒ　ｓｔｙｌｅ）プログラムを実行する。ＰＣＥ１４およびＣＣＥ１６は独立に動作しかつ組み合せて３２ビットのデータに対して１２００ＭＩＰより大きな処理能力を提供する。本発明の好ましい実施形態では、ＰＣＥ１４およびＣＣＥ１６は４ステージパイプライン構成でほぼ１００メガヘルツで動作する高性能３２ビットＲＩＳＣプロセッサで構成される。これらのＲＩＳＣプロセッサは、とりわけ、帯域内（ｉｎ－ｂａｎｄ）信号処理、エラー検出および訂正、およびチャネルプログラムによって規定される他のプロトコルおよびフォーマット処理のようなデータ処理のためにも使用することができる。

【００１７】ＰＣＰ１７はまたチャネルプログラムおよび／またはデータユニットを記憶するための暗号プロセッサＲＡＭ９（ＣＰ－ＲＡＭ）を含む。ＣＣ１１はデータユニットを処理する前にチャネルプログラムをＣＰ－ＲＡＭ９から処理エンジンのメモリへとダウンロードする。ＣＣ１１はまたデータユニットを処理する前にＣＰ－ＲＡＭ９から処理エンジンのメモリへとチャネルプログラムのコンテクストをダウンロードする。

【００１８】ＫＭＣＥ１２は、とりわけ、システム１０のためのマスタ制御機能を達成する。好ましい実施形態では、ＫＭＣＥ１２はＫＭＣＥ１２内のＲＯＭに組み込まれた保安オペレーティングシステム（ＳＯＳ）を含む。好ましい実施形態では、ＦＳ－ＲＩＳＣ２６は高性能３２ビットＲＩＳＣプロセッサである。ＦＳ－ＲＩＳＣ２６に加えて、ＫＭＣＥ１２は好ましくはパブリックキープログラムの処理に適した数値演算コプロセッサまたはマスコプロセッサ（ｍａｔｈ　ｃｏｐｒｏｃｅｓｓｏｒ）を含む。この実施形態では、ＫＭＣＥ１２は複数チャネルおよび単一チャネルの埋め込まれた（ｅｍｂｅｄｅｄ）アプリケーションの実行を可能にするためにおよそ１５０ＭＩＰの処理能力を有する。

【００１９】他の実施形態では、システム１０は種々のアプリケーションのために埋め込まれた暗号処理要素として作用することができる。例えば、システム１０はデータフロースルーアーキテクチャ（ｄａｔａ　ｆｌｏｗ　ｔｈｒｏｕｇｈ　ａｒｃｈｉｔｅｃｔｕｒｅｓ）またはコプロセッサアーキテクチャ（ｃｏｐｒｏｃｅｓｓｏｒ　ａｒｃｈｉｔｅｃｔｕｒｅ）が実施できるようにする。データフロースルーアーキテクチャにおいては、データは平文インタフェースポート３３から暗号文インタフェースポート３７へあるいはその逆に流れることができる。システム１０に組み込まれたまたは埋め込まれた内部保安メカニズムは微妙な（ｓｅｎｓｉｔｉｖｅ）平文データおよび変数のような論理的に異なるデータタイプを保護される暗号文データから隔離または分離することを助ける。コプロセッサアーキテクチャの構成では、例えばホストシステムが前記タイプまたは形式のデータを隔離するためにより大きな設計の確実さ（ｄｅｓｉｇｎ　ａｓｓｕｒａｎｃｅ）を好適に提供する。

【００２０】システム１０の好ましい実施形態では、ＰＴＩＰ１３およびＣＴＩＰ１５はＦＩＦＯ制御構造を備えてポート３３および３７において８ビット、１６ビットおよび３２ビット並列データインタフェースを含む。インタフェースプロセッサ１３および１５もまた好ましくは直列非同期および直列同期インタフェースを含む。ＰＴＩＰ１３およびＣＴＩＰ１５は内部プロセッサ、内部物理メモリおよび外部メモリ拡張能力を含む。インタフェースプロセッサのメモリはそれらの内部プロセッサによって管理される。好ましい実施形態では、インタフ

ェースプロセッサは全２重（ｆｕｌｌ　ｄｕｐｌｅｘ）動作が可能でありかつ平文および暗号文データを処理するために完全な物理的データインタフェースのアイソレーションを提供する。

【００２１】インタフェースポート３１はＫＭＣＥ１２と関連しており、かつ好ましくはメモリインタフェース、構成信号（ｃｏｎｆｉｇｕｒａｔｉｏｎ　ｓｉｇｎａｌｓ）、システムクロックおよび割込みのためのポートを含む。好ましい実施形態では、メモリインタフェースポートは３３ビットのデータバス、２４ビットのアドレスバスおよび内部メモリまたはＩ／Ｏ装置をアクセスするための制御インタフェースから構成される。システム１０の好ましい実施形態では、ＫＭＣＥ１２はＰＴＩＰ１３またはＣＴＩＰ１５を通してコマンドおよびデータを受ける。他の実施形態は制御およびデータがインタフェースポート３１からくるようにすることができる。

【００２２】システム１０はまたコンテクストポート（ｃｏｎｔｅｘｔ　ｐｏｒｔ）３６に接続するコンテクストメモリバス３８（ＣＮＴＸ）を含む。好ましい実施形態では、コンテクストメモリバス３８は外部コンテクストメモリに結合するために使用される３３ビットのデータバスおよびアドレス制御バスから構成される。ＣＣ１１はＰＣＰ１７におけるアクティブなタスクから外部コンテクストメモリにおけるインアクティブなタスクへのコンテクストの交換またはスワッピングを管理する。バス３８は内部メモリに存在し得るものよりも多くの同時的なタスクを要求するアプリケーションのための高速のコンテクスト変化を可能にする。ポート３２は制御信号および個別の警報信号のためにＣＣ１１へのインタフェースを提供する。

【００２３】ここで使用されているコンテクスト（Ｃｏｎｔｅｘｔ）は、例えば、特定のチャネルプログラムに関連する情報を含みかつ状態（ｓｔａｔｅ）または変数情報（ｖａｒｉａｂｌｅ　ｉｎｆｏｒｍａｔｉｏｎ）、キーおよびチャネルに関連する機能情報を含むことができる。

【００２４】好ましい実施形態では、本発明の暗号処理システムは超大規模集積回路（ＵＬＳＩ）装置において、好ましくは単一シリコンダイ上で、実施される。好ましい実施形態では、いくつかの処理サブシステムが前記ＵＬＳＩ内に集積され広範囲の暗号プログラムのクラスにとって適切なほぼ１３５０ＭＩＰの処理能力を得ることができる。

【００２５】図２は、本発明の好ましい実施形態に係わるデータユニットの処理を示す。本発明の暗号処理システムのアーキテクチャは非常に高いスループットを備えた複数チャネルのパケット化通信スレッド（ｔｈｒｅａｄｓ）の処理を可能にする。内部サブシステムと外部ホストとの間の非同期動作はＣＣ１１における有限状態マシン（ｆｉｎｉｔｅ　ｓｔａｔｅ　ｍａｃｈｉｎｅ）によって管理される。

【００２６】図２を参照すると、データユニット４１は、時間線またはタイムライン（ｔｉｍｅ－ｌｉｎｅ）４０で示されるように、外部ホストからインタフェースプロセッサ１３または１５（図１）の１つに転送される。インタフェースプロセッサはＣＣ１１に対し新しいデータユニット４１が処理のために用意ができていることを時間５１に該データユニットのヘッダをＣＣ１１に送ることにより通知する。データユニット４１のヘッダの情報に基づき、ＣＣ１１はインタフェースプロセッサにデータユニットを、ＫＭＣＥ１２、ＰＣＥ１４またはＣＣＥ１６のような、適切な処理サブシステムに移動することを指令する。好ましくは、データユニットの一部のみが、例えば、ヘッダ以外のすべてが、処理エンジンに転送される。

【００２７】データユニットがＰＣＰ１７におけるエンジンの１つによって処理されるべく準備されたとき、ＣＣ１１は処理を予定しかつ開始する。調停により、ＣＣ１１は好ましくはシステム１０における同時処理を最大にするためデータユニット転送を最大にする。タイムライン４２において、データユニット４１はＣＰ－ＲＡＭ９（図１）のようなメモリに転送され、そこで適切な処理エンジン（例えば、ＰＣＥ１４またはＣＣＥ１６）によって処理されるべく待機する。処理されるべき次のデータユニットのこのバックグランド作業はシステム１０にわたる潜伏（ｌａｔｅｎｃｙ）を最小にする。さらに、プログラムのバックグランド作業はＰＣＥ１４またはＣＣＥ１６の資源がデータユニットを処理しておりかつデータまたはプログラムを移動していないことを保証することを助ける。従って、システムのデータスループットが大幅に増大される。

【００２８】タイムライン４４は処理エンジンがデータユニット４５を処理しておりかつ処理されたデータ部分を出力インタフェースプロセッサに転送している期間を示す。時間フレーム５２は典型的には１クロックサイクルでありその間にキーおよびプログラムが切り替えられるコンテクスト切替え時間である。インタフェースプロセッサは時間５１においてＣＣ１１に対し新しいデータユニットが処理される用意ができたことを通知する。タイムライン４４の間に、処理されたデータユニットは処理ユニットから出力インタフェースプロセッサへと転送される。データユニットの処理は時間５４で完了する。この時間に、出力インタフェースプロセッサは外部ホストに対しデータユニットが処理を完了しておりかつ利用可能であることを通知する。データユニット４７は処理されたデータユニットであり、かつタイムライン４６の間に外部ホストに転送される。該データユニットを処理することに関連するパケットの潜伏５９は入力インタフェースプロセッサにおけるパケットの受信から処理されたデータユニットが外部ホストに転送される用意ができ

る時間までの時間として示されている。

【００２９】図２の処理ダイアグラムから見られるように、データユニットはパケット全体が処理ユニット（ＰＣＰ１７）によって受信された後に処理エンジン（例えば、ＰＣＥ１４またはＣＣＥ１６）によって処理される。さらに、データユニットはデータユニット全体が処理されるまで外部ホストに転送するために利用できない。データユニットは好ましくは複数のＤワード（Ｄｗｏｒｄｓ）（３２ビットのワード）からなり、その各々は個々に処理されかつ次に処理ユニットから出力インタフェースプロセッサへと処理が行われるに従って継続的な（ｃｏｎｔｉｎｕａｌ）ベースで送信される。好ましい実施形態では、外部ホストはデータユニット全体が処理を完了しかつ出力インタフェースプロセッサにおいて利用可能となった後に通知される。完全なデータユニットの処理は外部ホストからの行動またはアクションによって生じ得る行き詰まりまたはデッドロックを避ける働きをなす。

【００３０】出力インタフェースプロセッサは典型的にはデータユニットがそこから発出または発信される反対側のインタフェースポートに関連するインタフェースプロセッサである。例えば、プレインテキストまたは平文インタフェースポート３３において発出するデータは、それが処理された後に、ＣＴＩＰ１５に送られかつ暗号文インタフェースポート３７において利用可能にされる。

【００３１】好ましい実施形態では、データユニットはホストシステムによって非同期的にインタフェースプロセッサ１３または１５にロードされかつインタフェースプロセッサによって管理される。ＰＣＥ１４またはＣＣＥ１６による実行のために計画または予定されたデータユニットは処理ユニット（例えば、ＣＰ－ＲＡＭ９）に関連するメモリに送られかつ記憶される。処理エンジンがＦＳ－ＲＩＳＣ２６である場合は、処理の用意ができたデータユニットはメモリ２５に記憶される。インタフェースプロセッサ１３および１５はデータユニットの分解（ｄａｔａｕｎｉｔ　ｐａｒｓｉｎｇ）、優先順位付け（ｐｒｉｏｒｉｔｉｚｉｎｇ）、並列－直列および直列－並列変換、パケット統合、検査またはチェックワード（ｃｈｅｃｋｗｏｒｄ）発生およびメモリ管理機能のような機能を行う。

【００３２】本発明の好ましい実施形態では、システム１０によって処理されるデータユニットはシステム１０による処理のために特別にフォーマットされる。この実施形態では、インタフェースプロセッサ１３および１５は以下に説明するＡＰＤＵフォーマットでデータを処理する。しかしながら、ＡＰＤＵフォーマットにないストリームデータ（ｓｔｒｅａｍ　ｄａｔａ）もインタフェースプロセッサの並列または直列ポートにおいて受信されかつ処理のためにＡＰＤＵフォーマットへと変換することができる。

【００３３】図３は、本発明の好ましい実施形態と共に使用するのに適したデータユニットのフォーマットを示す。ＡＰＤＵフォーマットにおけるデータユニットが図３に示されている。ＡＰＤＵフォーマットのデータユニットは一連のＤワードからなる。各々のＤワードは欄６０に示されるオフセットを有する。最初のＤワードはチャネルヘッダのＤワード６６であり、これは好ましくは３２ビットのＤワードである。チャネルヘッダのＤワード６６に続いてコマンドＤワード６７があり、これは１のＤワードオフセットを有する。コマンドＤワード６７に続いて２と４０９４の間のＤワードオフセットを有するパラメータデータフィールド６８がある。ＡＰＤＵのパラメータデータフィールド６８はアプリケーションのペイロード（ａｐｐｌｉｃａｔｉｏｎ　ｐａｙｌｏａｄ）を含む。フィールド６８のデータは各々のチャネルに対して異なるフォーマットを持つことができ、それはアプリケーションプログラムは各々のデータユニットに対してコンテクスト交換（ｃｏｎｔｅｘｔ　ｓｗａｐ）できるからである。例えば、複数のチャネルに対してシステム１０において複数のプログラムが実行しているとき、いくつかのチャネルは通信スレッド（ｃｏｍｍｕｎｉｃａｔｉｏｎ　ｔｈｒｅａｄ）に対するロックステップ処理を保証するためにより堅牢な（ｒｏｂｕｓｔ）プロトコルを必要とするであろう。

【００３４】最後のＤワードは検査合計またはチェックサム（ｃｈｅｃｋ　ｓｕｍ）Ｄワード６９であり、これは好ましくはＡＰＤＵ全体にわたり計算される３２ビットのフレーム検査シーケンス（ｆｒａｍｅ　ｃｈｅｃｋ　ｓｅｑｕｅｎｃｅ：ＦＣＳ）である。検査合計またはチェックサム（ＣＳ）Ｄワード６９は特定の用途に対してイネーブルしまたはディスエーブルすることができる任意選択的なフィールドである。スタートアップ手順の間に、ＫＭＣＥ１２はＣＣ１１を構成しかつチェックサムが各々のＡＰＤＵに添付されたか否かを判定する。

【００３５】１つの適切なＦＣＳプログラムはＩＳＯ３３０９－１９６４Ｅ仕様の３２ビットのバージョンである。この仕様は情報処理システムおよびデータ通信のための高レベルデータリンク制御手順およびフレーム構造を規定する。

【００３６】図４は、本発明の好ましい実施形態において使用するのに適したチャネルヘッダのフォーマットを示す。該チャネルヘッダのフォーマットはチャネルヘッダのＤワード６６におけるフィールドのサイズまたは大きさおよびロケーションまたは位置を規定する。チャネルヘッダＤワード６６は３ビットのＡＰＤＵタイプフィールド７１、１９ビットのチャネルインデクスフィールド７２、１２ビットのＰＤＵ長さフィールド７３、スペアビット７４、３ビットのＭＬＳタグフィールド７５、優先度ビット７６およびパリティビット７７を含む。Ｍ

ＬＳタグフィールド７５および優先度ビット７６は任意選択的なものである。ＡＰＤＵタイプフィールド７１はＡＰＤＵタイプに対する値およびその対応する意義または有意性（ｓｉｇｎｉｆｉｃａｎｃｅ）を規定する。好ましくは、ＡＰＤＵタイプフィールド７１は、例えば、ＰＴＩＰ１３またはＣＴＩＰ１５からの、あるいはシステム１０の他の内部ソースからのＡＰＤＵのソースを規定する。ＡＰＤＵタイプフィールド７１は好ましくはデータユニットを受けるべき出力プロセッサをも示す。

【００３７】ＡＰＤＵタイプフィールド７１はまたＡＰＤＵが要求ＡＰＤＵであるか応答ＡＰＤＵであるかを特定する。応答ＡＰＤＵに対しては、チャネルインデクスフィールド７２はもはやチャネルインデクスを含まず、代わりに要求ＡＰＤＵのコマンドＤワードにおいて与えられる３ビットの要求プログラム番号（ｒｅｑｕｅｓｔ　ｐｒｏｇｒａｍ　ｎｕｍｂｅｒ：ＲＰＮ）を含む。ＣＣ１１はＡＰＤＵタイプフィールド７１を用いて、とりわけ、チャネルインデクスフィールド７２の使用（ｕｓｅ）を決定する。

【００３８】チャネルインデクスフィールド７２はデータユニットが通常のチャネルを呼んでいるか否かあるいはデータユニットが内部資源を呼んでいるかを規定する。例えば、チャネルインデクスの最初のビットが“１”であれば、最後の１０ビットは後に説明するチャネルテーブルにおいて使用するチャネルプログラムを識別する。チャネルテーブルはチャネルの特性を特定する。ＣＣ１１はコンテクストおよびプログラムが実行ユニットのアクティブチャネルメモリへと移動されかつアクティブチャネルメモリから出される際にチャネルテーブルを管理する。チャネルが生成されるとき、エントリが該チャネルテーブルに加えられる。チャネルテーブルのエントリが除去されたとき、そのチャネルはインアクティブ（ｉｎａｃｔｉｖｅ）になる。インアクティブなチャネルのテーブルは状態（ｓｔａｔｅ）および変数データおよび／またはプログラムがＣ１１の状態マシンによってアクセスできない記憶位置に移動されたものである。ＦＳ－ＲＩＳＣ２６上で動作するアプリケーションプログラムは該テーブルからチャネルプログラムを再割当てしかつデータをＰＣＰ１７から除去することができる。インアクティブなチャネルデータを記憶するために使用されるメモリはＫＭＣＥ１２または外部コンテクストメモリ内に設けることができる。

【００３９】チャネルインデクスフィールド７２に関しては、もしチャネルインデクスの最初のビットがゼロであれば、データユニットは処理のために内部資源を要求しているかもしれない。このチャネルインデクスの次の１０ビットはどの内部資源が要求されているかを示す。内部資源はＰＴＩＰ１３、ＣＴＩＰ１５、ＣＣ１１、ＰＩ２１内のランダマイザ（ｒａｎｄｏｍｉｚｅｒ）、およびＦＳ－ＲＩＳＣ２６を含む。

【００４０】ＰＤＵ長さフィールド７３は好ましくはコマンドＤワード６７に続く任意選択的なＣＳスペースＤワードを含むＤワードの数を示す。長さフィールド７３はアプリケーションデータのサイズまたは大きさを規定する。図３に示される実施形態では、最大のアプリケーションデータユニットのサイズは４０９４Ｄワードであり、これは１３１，００８ビットである。

【００４１】ＭＬＳタグフィールド７５はＡＰＤＵの保安レベルを特定する。好ましい実施形態では、ＭＬＳタグフィールド７５内の値はチャネルに関連するキーのＭＬＳタグの値と比較される。２つのタグが整合しない場合、データユニットは排除されかつエラー状態がセットされる。本発明の好ましい実施形態では、キーのＭＬＳタグは該キーと共にロードされるかあるいはキー作成のときに特定される。該キーのＭＬＳタグは好ましくは該キーを作成するために使用される保安レベルに基づく。

【００４２】優先度ビット７６はＡＰＤＵのための優先度レベルを規定する。該優先度ビットは好ましくはインタフェースプロセッサ１３または１５によって使用されてデータユニットの処理の順序を選択する。示された実施形態では、２つのレベルの優先度がある。例えば、ゼロは非実時間（ｎｏｎ－ｒｅａｌ－ｔｉｍｅ）処理を特定し、一方“１”は実時間処理を特定するために使用される。

【００４３】パリティビット７７は好ましくはそれぞれのヘッダＤワードに加えられる。ＣＣ１１は該ヘッダワードのパリティを該ヘッダが処理されるときに検査する。

【００４４】図５は、本発明の好ましい実施形態において使用するのに適したコマンドＤワードのフォーマットを示す。コマンドＤワード６７は好ましくは各々のＡＰＤＵにおける第２のＤワードである。コマンドＤワード６７は１０ビットのコマンドＩＤフィールド８１、７ビットの応答フィールド８２、３ビットの要求プログラム番号（ＲＰＮ）フィールド８３、５ビットのＡＰＤＵ長さフィールド８４、スペアビット８５、およびパリティビット８６を含む。好ましい実施形態では、コマンドＩＤフィールド８１はデータユニットに対して実行されるべき機能を特定する。機能は好ましくは各々のチャネルプログラムに対して規定される。好ましい実施形態では、システム１０に対して本来の機能はない。機能は、例えば、暗号化、暗号解読、符号、真正証明、その他を含むことができる。例えば、暗号化のような機能はアプリケーションソフトウエアに対しＡＰＤＵのデータ部分（例えば、パラメータデータフィールド６８）が暗号化されるべきことを指定する。該暗号化はチャネルプログラムおよびチャネルインデクスフィールドによって選択されたそのチャネルに対して特定されたキーを使用して行われる。

【００４５】応答フィールド８２は処理されたデータユ

ニットと共に処理ステータスを戻す。該応答はシステム１０の処理ユニットによって発生される。例えば、ＰＣＥ１４は出力インタフェースプロセッサへのデータユニットの送信の終りに「処理完了（ｐｒｏｃｅｓｓｉｎｇ　ｃｏｍｐｌｅｔｅ）」応答を応答フィールド８２に提供することができる。同様に、ＣＣ１１はデータユニットの転送が失敗した場合に送信プロセッサに「デフォールト」応答値を送ることができる。応答フィールド８２は特定のアプリケーションまたはチャネルプログラムに依存するものとすることができる。

【００４６】ＲＰＮフィールド８３は要求形式の（ｒｅｑｕｅｓｔ　ｔｙｐｅ）ＡＰＤＵにおいてどのプログラムが要求を発行したかを識別するために使用される。ＣＣ１１は、例えば、暗号化エンジンの１つで現在動作している処理の１つにマッピングするためにＲＰＮを使用することができる。ＲＰＮフィールド８３はＣＣ１１がＡＰＤＵを正しいプロセッサに導くことができるようにする応答ＡＰＤＵチャネルインデクスにおける値を戻す。好ましい実施形態では、ＡＰＤＵが外部ホストから発出した場合、ＲＰＮフィールドは使用されずかつゼロにセットされる。プログラムを識別することにより、ＲＰＮフィールド８３はシステム１０の実行ユニットにおいて動作している異なるチャネルプログラムの間でコマンド、パラメータおよびデータを要求しかつ受け渡す。処理ユニットはプログラムを同時に走らせることができるから、処理ユニットはまたデータユニットを通信構造で使用する。従って、ＲＰＮフィールド８３の使用によって、プログラムは情報をＣＣ１１を使用してそれら自体の間で転送できる。

【００４７】ＡＰＤＵ長さフィールド８４はＡＰＤＵのサイズを規定する。パリティビット８６がコマンドＤワード６７のヘッダに加えられる。ＣＣ１１は該ヘッダワードに関するパリティをそれがコマンドＤワード６７（図３）を処理するときに検査することができる。

【００４８】図６は、本発明の好ましい実施形態において使用するのに適したチャネル規定またはチャネル定義テーブルを示す。ヘッダＤワード６６（図４）のチャネルインデクスフィールド７２（図４）はＣＣ１１（図１）によって読み取られてＡＰＤＵに適用されるチャネルテーブル９０の行（ｒｏｗ）を決定する。チャネルテーブル９０はチャネルテーブルフィールドの内容およびそれらの長さを規定する。好ましい実施形態では、チャネルテーブル９０は各々のチャネルを特性付けるために１０２４ワード長さ×３２ビットのテーブルとされる。ＣＣ１１は処理エンジン１４および１６においてチャネルプログラムをセットアップする場合にチャネルテーブル９０のフィールドを使用する。チャネルテーブル９０は２ビットの割り当てられた活動フィールド（ａｌｌｏｃａｔｅｄ　ａｃｔｉｖｉｔｙ　ｆｉｅｌｄ）９２、要求サービスビット９３、セイブバックビット（ｓａｖｅ　ｂａｃｋ　ｂｉｔ）９４、４ビットのプログラムＩＤフィールド９５、１７ビットの可変アドレスフィールド９６、４ビットの長さフィールド９７、および3ビットのＭＬＳキータグフィールド９８を含む。

【００４９】チャネルテーブル９０の情報はＡＰＤＵを適切な処理資源に導くために使用されかつそのチャネルに対する特定の通信スレッドのインストールまたは再インストールのための他の情報を含む。一般に、チャネルテーブルはチャネル定義または規定のためにプログラムおよびコンテクストが配置されるロケーションへのポインタを含む。チャネルインデクスフィールド７２もまたＰＣＰ１７に割り当てられていないチャネルに対し指示する（ｐｏｉｎｔ）ことができる。この場合、ＣＣ１１は処理が行われるＫＭＣＥ１２へとデータユニットを導くことができる。一般に、処理はＫＭＣＥ１２によって例外ベース（ｅｘｃｅｐｔｉｏｎ　ｂａｓｉｓ）で行われる。

【００５０】チャネルインデクスフィールド７２はＦＳ－ＲＩＳＣ２６上で実行するアプリケーションプログラムのソフトウエアによって割り当てられかつチャネルの作成／規定時に生じる。チャネルインデクスの割当ては特定のアプリケーションに依存して固定されあるいは動的なものとされる。動的チャネル割当てはチャネル作成のときにおける値の交換を含みそれによって、例えば、外部ホストが適切にＡＰＤＵを構築できるようにする。新しいチャネルが作成されあるいは取りこわされた（ｔｏｒｎ　ｄｏｗｎ）とき、ＫＭＣＥ１２はチャネルテーブル９０において新しいエントリを作成しあるいはエントリを削除する。好ましい実施形態におけるチャネルテーブル９０はＣＣ１１のメモリ１９に格納される。

【００５１】各々のチャネルはＣＣ１１内に内在的に格納される関連するチャネル状態（ｃｈａｎｎｅｌ　ｓｔａｔｅ）を有する。チャネル状態は動作している現在のプログラム状態、次のまたは最後の状態、スタンバイ状態、インストール状態およびインアクティブまたは不活性状態を含む。チャネルは現在のプログラム状態およびコンテクストがＰＣＥ１４またはＣＣＥ１６上で実行している場合に動作状態（ｒｕｎｎｉｎｇ　ｓｔａｔｅ）にある。好ましい実施形態では、ＰＣＥ１４およびＣＣＥ１６は一組の少なくとも４つのメモリを有し、これらはピンポン（ｐｉｎｇ－ｐｏｎｇ）様式で選択され現在のチャネルが現在実行している間に次のチャネルがロードできるようにする。前記メモリ規定または定義はこのメモリスワッピングが発生するからアクティブ（ａｃｔｉｖｅ）からシャドウ（ｓｈａｄｐｗ）へと変化する。

【００５２】次のまたは最後のチャネル状態はチャネルプログラムが上に述べたＰＣＥ１４またはＣＣＥ１６と関連するシャドウメモリに存在することを示す。スタンバイチャネル状態はアプリケーションプログラムがＣＰ－ＲＡＭ９に存在しかつシャドウメモリにインストール

されるべく用意ができていることを規定する。インストールチャネル状態は、１つのチャネルに対するチャネルプログラムに関連するコンテクストが他のものとスワップされたときに、スタンバイと次のまたは最後のものとの間のチャネル状態である。インアクティブチャネル状態は前記コンテクストおよび／またはプログラムがＰＣＰ１７の制御の外にある状態である。例えば、プログラムはＫＭＣＥ１２にあるいは外部コンテクストメモリに存在することができる。

【００５３】いったんチャネルがＰＣＰ１７において確立されると、外部ホストのアプリケーションはＦＳ－ＲＩＳＣ２６において実行しているアプリケーションプログラムからの介在なしにチャネルごとのベースでＰＣＰ１７においてＡＰＤＵを処理することができる。従って、暗号化または暗号解読のような機能の間の最大のスループットがＰＣＰ１７におけるチャネルごとの（ｐｅｒ　ｃｈａｎｎｅｌ）自律的処理によって達成される。従って、典型的なアプリケーションはＫＭＣＥ１２の介在なしにＡＰＤＵをＰＣＰ１７を通って受け渡す。

【００５４】チャネル活動フィールド９２は有効なチャネルを識別し、かつチャネル状態情報を含む。無効なチャネルプログラムを識別するデータユニットは処理のためにＦＳ－ＲＩＳＣ２６に書き込まれることができる。チャネル活動フィールド９２は、有効な場合、そのチャネルの処理活動を示す。データユニットが処理されている場合、チャネル活動フィールド９２は更新される。チャネル活動フィールド９２はまたＫＭＣＥ１２によって使用されてどのチャネルがよりまれにしか使用されていないかかつ従って、システム１０への最小の影響と共に除去できるかを決定する。好ましい実施形態では、チャネル活動フィールド９２は順次、例えば、番号“０１”．“１０”および“１１”を通って更新される。現在のまたは現行の値は特定のＡＰＤＵを処理するために使用されているチャネルと共に記憶される。チャネル活動フィールド９２の値はチャネルの最後に使用された状態を表す。

【００５５】要求サービスビットフィールド９３はＰＣＥ１４またはＣＣＥ１６上で実行しているアプリケーションプログラムがＦＳ－ＲＩＳＣ２６上で実行しているアプリケーションプログラムによって更新された包括的変数（ｇｌｏｂａｌ　ｖａｒｉａｂｌｅｓ）に対して新しい値を読む必要があることを示す。従って、要求サービスビットまたはサービス要求ビットがセットされている場合、ＦＳ－ＲＩＳＣ２６はプログラムが開始する前に付加的な情報を提供する。セイブバックビット９４はＣＣ１１がコンテクストをセーブするために使用する方法を示す。例えば、コンテクストはＣＰ－ＲＡＭ９にあるいは外部メモリにセーブすることができる。コンテクストは一般に処理エンジンの１つにおいて導入されたまたはインストールされたチャネルプログラムが現存するコンテクストのいくつかまたはすべてを変えた後にセーブし戻される。セーブバックビット９４を使用することにより、多くのＡＰＤＵは同じチャネル上で動作してコンテクストへの変化を生じさせることができる。チャネルが処理エンジンの１つから除去されたとき、コンテクストは内部または外部メモリへとセイブバックされる。従って、無用のセーブが避けられる。

【００５６】プログラムＩＤフィールド９５はアルゴリズムまたはプログラムのためのＩＤコードを含む。好ましくは、プログラムＩＤフィールド９５はＣＣ１１のメモリ１９内に存在するプログラムアドレステーブルにおける行（ｒｏｗ）を指示する。プログラムアドレステーブルは、とりわけ、ＣＣ１１が異なるプログラムを追跡するために使用するフィールドを規定する。変数または可変アドレスフィールド９６はチャネルに対する可変または変数データが位置するＰＣＰ１７内のメモリロケーションの開始アドレスを規定する。可変または変数アドレスフィールド９６を使用することにより、ＣＣ１１は該変数が現在アクティブメモリにあるか、シャドウメモリにあるか、変数または状態メモリにあるかを決定する。さらに、変数アドレスフィールド９６はＣＣ１１に対し変数データがＣＰ－ＲＡＭ９にありかつ変数状態に対してＣＣＥ１６のＰＣＥ１４のアクティブまたはシャドウメモリへと移動されるべきことを示す。シャドウおよびアクティブメモリに対するメモリアドレスは好ましくは固定され従ってＣＣ１１がＡＰＤＵが実行の用意ができているか否かあるいはＡＰＤＵがシャドウメモリへとスタグされるべきか否かを決定できるようにする。ＡＰＤＵがシャドウメモリにスタグされている間に、プログラム変数および状態のような、そのチャネルに対するチャネルパラメータが処理エンジンにロードされる。

【００５７】状態長さフィールド９７は上で述べた状態変数データの長さを規定する。好ましい実施形態では、状態長さフィールド９７はゼロと３２のＤワードとの間で変動する。ＭＬＳキータグフィールド９８はチャネルキーの保安レベルを列挙する（ｌｉｓｔｓ）。キータグフィールド９８の値はＡＰＤＵのヘッダＤワード６６のＭＬＳタグフィールド７５において受信されたタグと比較される。キータグフィールド９８に列挙されたチャネルキーの保安レベルは処理されるべきデータユニットに対してヘッダＤワード６６のＭＬＳタグフィールド７６において識別される、データの保安レベルより高くあるべきである。

【００５８】図７は、本発明の好ましい実施形態において使用するのに適したプログラムアドレステーブルの例を示す。プログラムアドレステーブル７００はプログラムタイプフィールド７０２、プログラムロケーションフィールド７０３、赤／黒（ｒｅｄ／ｂｌａｃｋ）フィールド７０４、プログラムアドレスフィールド７０５、プログラム長さフィールド７０６、ブランクＤワードフィ

ールド７０７および変数長さフィールド７０８を含む。チャネルテーブル９０からのプログラムＩＤフィールド９５（図６）はチャネルアドレステーブル７００の行（ｒｏｗ）を指示する。従って、各々のチャネルはプログラムアドレステーブル７００における行と関連している。

【００５９】プログラムタイプフィールド７０２はチャネルプログラムのサイズ、例えば、大きいかまたは小さいか、を識別する２ビットのフィールドである。プログラムタイプフィールド７０２はまたプログラムが動作する実行ユニット、例えば、ＰＣＥ１４またはＣＣＥ１６、を識別する。プログラムロケーションフィールド７０３はそのチャネルに対するチャネルプログラムのロケーションを識別する。ＣＣ１１はプログラムロケーションフィールド７０３を使用してそれがＡＰＤＵを処理するために必要な場合にプログラムのロケーションを決定する。プログラムロケーションフィールド７０３はまたプログラムがいつＦＳ－ＲＩＳＣ２６によってロードされるべきかを示す。プログラムロケーションフィールドはまたそのプログラムの１つのコピーのみが存在しかつそれがＰＣＥ１４またはＣＣＥ１６のような処理エンジンに永久的に存在することを示すことができる。プログラムロケーションフィールド７０３はまたはチャネルプログラムがＣＰ－ＲＡＭ９にありかつ適切な処理エンジンにとって必要な場合にコピーされるべきことを示す。プログラムロケーションフィールド７０３はまたプログラムが外部メモリにありかつ必要に応じてシステム１０にコピーされることを示す。プログラムが外部メモリにあるとき、該プログラムは処理エンジンの１つにおけるインストールの前にＫＭＣＥ１２によって解読される必要があるかもしれない。

【００６０】赤／黒フィールド７０４は好ましくはプログラムの保安レベルを識別する２ビットのフィールドである。赤／黒フィールド７０４は、とりわけ、該プログラムが保安プログラムであるかあるいは保安プログラムでないかを示す。保安プログラムは暗黒のまたはブラック（ｂｌａｃｋ）外部メモリに移動される前に暗号化され、かつ外部メモリから該プログラムを移動した後に解読されるべきである。非保安プログラムは外部メモリに移動しまたは外部メモリから移動する前に暗号化される必要はない。本発明の好ましい実施形態では、赤／黒フィールド７０２は外部メモリがプログラム記憶のために使用されないかあるいはプログラム記憶のために利用できない場合には使用されない。

【００６１】プログラムアドレスフィールド７０５はチャネルプログラムまたはそのチャネルに対するプログラムのメモリロケーションを識別するアドレスポインタを含む。該メモリロケーションはＰＣＥ１４、ＣＣＥ１６、ＣＰ－ＲＡＭ９または外部メモリにあることができる。ＣＣ１１はプログラムアドレスフィールド７０５を使用してチャネルプログラムの位置を決定しかつそれを処理エンジンのシャドウメモリ内に移動する。特定のプログラムが処理エンジンに永久的にロードされる場合は、前記プログラムアドレスはプログラムが移動する必要がないことを示すためにある値を含むことができる。

【００６２】プログラム長さフィールド７０６はメモリに記憶されるチャネルプログラムのマイクロコードのサイズを識別する。ブランクＤワードフィールド７０７はＣＣ１１がそのメモリにプログラムをインストールした後にＣＣ１１がメモリロケーションに書き込むゼロまたはブランクＤワードの数を示す。前記ゼロまたはブランクＤワードはプログラムスペースがオーバライトされていることを保証するために前にインストールされたプログラムに続き処理エンジンのプログラムスペースに書き込まれる。

【００６３】変数長さフィールド７０８はこの特定のプログラムにおいて使用される変数の長さを含む。変数の長さは同じプログラムを使用するすべてのチャネルに対して同じとすることができる。プログラム変数の長さは好ましくはゼロと３２Ｄワードの間である。ＣＣ１１はチャネルのコンテクストを処理エンジンにインストールする場合に前記変数長さを使用する。

【００６４】図８は、本発明の好ましい実施形態において使用するのに適したセットアップおよび構成（ｃｏｎｆｉｇｕｒａｔｉｏｎ）手順のフローチャートである。手順１００はシステム１０によって、とりわけ、チャネルを定義または規定し、かつ関連するチャネルプログラムをＰＣＰ１７にロードするために行われる。本発明のプログラム可能暗号処理システムは、そのスーパースカラ（ｓｕｐｅｒｓｃａｌｅｒ）プログラム可能アーキテクチャにより、同時に動作するいくつかのプログラムをもつことができる。これらのプログラムはＦＳ－ＲＩＳＣ２６の保安オペレーティングシステム上で動作するマスタアプリケーションプログラムからインストールされる。タスク１０２において、ＫＭＣＥ１２はシステム１０の構成要素（ｃｏｍｐｏｎｅｎｔｓ）およびサブシステムが適切に動作していることを保証するためリセットおよびセルフテスト処理を行う。タスク１０４においては、マスタアプリケーションプログラムが外部ホスト１０３からＫＭＣＥ１２へとロードされる。本発明の別の実施形態では、アプリケーションプログラムはＫＭＣＥ１２のメモリ２５内に存在し、かつメモリ２５からＦＳ－ＲＩＳＣ２６へとロードされる。

【００６５】タスク１０６においては、タスク１０４においてロードされたアプリケーションプログラムが実行され、好ましくはＦＳ－ＲＩＳＣ２６の保安オペレーティングシステム上で実行される。

【００６６】タスク１０８においては、アプリケーションプログラムはＣＣ１１に対しチャネル定義情報１０７を使用して複数のチャネルを作成しかつ定義するよう指

令する。チャネル定義情報またはチャネル規定情報（Ｃｈａｎｎｅｌ　ｄｅｆｉｎｉｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎ）１０７はシステム１０内に記憶されあるいは外部ホストによって提供することができる。このステップの間に、チャネルテーブル９０（図６）のようなチャネルテーブルが作成される。さらに、図７のプログラムアドレステーブル７００のようなプログラムアドレステーブルも作成される。本発明の好ましい実施形態では、これらのテーブルは共有または共用メモリ１８（図１）に記憶される。セットアップおよび構成手順１００のタスク１０８の間に、チャネルプログラムは好ましくは処理エンジン１４または１６においてインストールされない。チャネルプログラムはデータユニットが処理されるときに特定のデータユニットのためにインストールされる。例えば、ＡＰＤＵのチャネルインデクスはＣＣ１１にチャネルプログラムが走る（ラン）ようにさせかつＣＣ１１はこのプログラムをインストールしかつ該プログラムの実行を適切な処理エンジンにおいて開始する。

【００６７】チャネル定義情報１０７はコンテクストの特定のプログラムまたはプログラムセグメントとの関連を定義するまたは規定する情報を含む。実行コード（ｅｘｅｃｕｔｉｏｎ　ｃｏｄｅ）の単一のスレッド（ｔｈｒｅａｄ）はチャネルの例である。コンテクストを交換するマルチ処理システムにおいては、数多くの同時的なチャネルがマルチチャネル動作のために時分割で動作する。従って、各々のチャネルに対する別個のコンテクストが好ましくは維持される。

【００６８】チャネルが定義されかつチャネルプログラムが、タスク１１０において、識別された後、アプリケーションプログラムは特定のチャネルプログラムをＰＣＰ１７のＣＰ－ＲＡＭ９のようなメモリにダウンロードする。好ましくは各々のチャネルに関連してチャネルプログラムがある。

【００６９】タスク１１２においては、暗号キー（ｅｎｃｒｙｐｔｉｏｎ　ｋｅｙｓ）がシステム１０にロードされる。好ましくは、該キーはフィルポート（ｆｉｌｌ　ｐｏｒｔ）３４を通してプログラム可能インタフェース２１へとロードされる。キーは、暗号化、暗号解読、デジタル署名および真正証明のために使用されるキーを含めて、ＤＥＳ暗号キー、パブリックおよびプライベートキーおよび暗号法の技術においてよく知られた他の形式のキーを含む。好ましい実施形態では、メモリ２５はシステム１０への電源障害の場合にキーの喪失を防止するため、バッテリのような、バックアップ電源を有する。タスク１１２は任意選択的にＦＳ－ＲＩＳＣ２６においてキーを発生するキー発生タスク１１１を含むことができる。ＦＳ－ＲＩＳＣ２６によって行われるキー発生はパブリックまたはプライベートキー発生ソフトウエアを使用することを含むことができる。ＦＳ－ＲＩＳＣ２６は内部ランダマイザ（ｒａｎｄｏｍｉｚｅｒ）を使用することによるものを含めて技術的に知られた多くの方法でチャネルまたはセッションキーを発生することができる。好ましい実施形態では、キーはチャネルに関連され、かつチャネルを適切なキーまたはキー対と関連させるテーブルに記憶される。好ましい実施形態では、キーはＦＳ－ＲＩＳＣ２６の初期化の間に各チャネルと関連される。１実施形態では、チャネルの保安レベルはそれを特定のキーと関連させる。

【００７０】もしタスク１１２がキー発生タスク１１１を含んでいれば、キーはキーエスクロー（ｋｅｙ　ｅｓｃｒｏｗ）にとって利用できるものとすることができる。タスク１１２はまたキーをキーエスクローに提供するタスクを含むことができる。タスク１１４においては、キーはチャネル情報１１３を使用するチャネルと関連されかつデータユニットを処理する上で使用するためにＣＰ－ＲＡＭ９またはＰＣＥ１４またはＣＣＥ１６と関連するローカルメモリのような、メモリに記憶される。タスク１１４が完了すると、システム１０はデータユニットを処理する用意ができている。

【００７１】図９は、本発明の好ましい実施形態において使用するのに適したデータユニットの処理手順のフローチャートである。好ましい実施形態では、手順２００はシステム１０によって受信された各々のデータユニットに対して行われる。一般に、手順２００はある機能を各々のデータユニットに対して行われるようにする。機能は、例えば、暗号化、暗号解読、署名または真正証明を含む。該機能が行われかつ処理されたデータユニットが完成した後、システム１０は処理されたデータユニットを外部ホストにとって利用できるようにする。

【００７２】タスク２０２においては、データユニットは外部ホストからインタフェースプロセッサ１３または１５において受信される。データユニットは好ましくは図３～図５で説明したようなＡＰＤＵフォーマットになっている。データユニットは外部ホストによって他のフォーマットからＡＰＤＵフォーマットへと変換できる。例えば、ＡＰＤＵフォーマットでないストリームデータ（ｓｔｒｅａｍ　ｄａｔａ）の場合は、ＰＴＩＰ１３またはＳＴＩＰ１５は該ストリームデータをシステム１０に記憶された構成情報２０３を使用してフォーマットすることができる。好ましい実施形態では、外部ホストはタスク２０２においてインタフェースプロセッサによって受信される前にデータをＡＰＤＵフォーマットに変換するが、システム１０がデータユニットをＡＰＤＵフォーマットに変換することを妨げるものは何もない。

【００７３】構成情報（Ｃｏｎｆｉｇｕｒａｔｉｏｎ　ｉｎｆｏｒｍａｔｉｏｎ）２０３はシステムのアプリケーションに基づく特定の情報を含む。例えば、構成情報２０３は処理されるべきデータユニットの種別、使用されるべきインタフェース、ＡＰＤＵフォーマット情報、およびいつＰＴＩＰ１３またはＣＴＩＰ１５がＡＰＤＵ

を生成するかを含むことができる。

【００７４】タスク２０２はデータユニットを同期的にあるいは好ましくは非同期的に受信することができる。データユニットはまた関連するインタフェースプロセッサの並列または直列ポートを通して並列または直列形式で受信することができる。データユニットが非同期的に受信される場合、インタフェースプロセッサは外部ホストにそれがデータユニットを受信するために利用できることを通知する。

【００７５】タスク２０４においては、入力インタフェースプロセッサがパケット優先度（すなわち、ヘッダＤワード６６（図４）のビット７６）を読み取り、かつそのデータユニットに対する処理を予定または計画する。好ましい実施形態では、実時間優先度を備えたパケットが始めにＣＣ１１に送られ、それに続き非実時間パケットが送られる。タスク２０４の一部として、インタフェースプロセッサはＣＣ１１に対し新しいデータユニットが処理される用意ができていることを通知する。タスク２０６においては、ＣＣ１１はデータユニットのヘッダを読み取る。

【００７６】タスク２０８においては、ＣＣ１１はフィールド７２からチャネルインデクスを、フィールド７１からＡＰＤＵタイプを、そしてデータユニットのヘッダＤワード６６のＭＬＳタグフィールド７５を読み取り該データユニットを処理するために適切なチャネルプログラムおよび処理資源を決定する。ＣＣ１１はまたタスク２０８の一部としてＡＰＤＵ長さフィールド８４を読み取ることができる。

【００７７】タスク２１０においては、ＣＣ１１はインタフェースプロセッサに対しデータユニットを、ＰＣＥ１４またはＣＣＥ１６のような処理エンジンあるいはＦＳ－ＲＩＳＣ２６に導くよう指令する。処理エンジンはタスク２０８からの情報に基づき選択される。ヘッダＤワード６６のチャネルインデクスフィールド７２はデータユニットに対して処理を行うために外部ユニットを決定する。好ましい実施形態では、データユニットはＣＰ－ＲＡＭ９に導かれ、そこでＰＳＥ１４またはＣＣＥ１６による処理を待つ。あるいは、データユニットはＰＣＥ１４またはＣＣＥ１６のシャドウメモリに導かれそこでそれぞれＰＣＥ１４またはＣＣＥ１６による処理を待つ。

【００７８】タスク２１０の間に、ＣＣ１１におけるフレームチェックシーケンス（ｆｒａｍｅ　ｃｈｅｃｋ　ｓｅｑｕｅｎｃｅ：ＦＣＳ）チェッカが転送の間のデータユニットの完全性を調べる。ＦＣＳによって問題が発生したとき、デフォールト応答がデータユニットを提供した外部ホストに戻される。ヘッダＤワード６６のＰＤＵ長さフィールド７３はＰＣＰ１７においてメモリを割り当てるためにＣＣ１１によって使用される。本発明の１実施形態では、タスク２１０はデータユニットのペイロード部分のみを処理エンジンに導くことを含む。

【００７９】タスク２１１においては、その特定のチャネルに対するコンテクストがダウンロードされる。タスク２１２においては、ＣＣ１１はチャネルプログラムが適切な処理エンジンにダウンロードされるようにする。好ましくは、プログラムはＰＣＥ１４またはＣＣＥ１６（図１）のシャドウメモリにロードされる。

【００８０】タスク２１４においては、ＭＬＳタグフィールド７５がチャネルテーブルにおけるタグ（すなわち、ＭＬＳキータグフィールド９８）と比較されてプログラムの保安レベルが少なくともそのデータユニットが要求する保安レベル程度に大きいことを保証する。データユニットがチャネルが提供するより大きな保安性を要求する場合は、該データユニットは好ましくは処理されずかつデフォールト応答が入力インタフェースプロセッサに戻される。インタフェースプロセッサはこのデフォールト応答をそのデータユニットを提供した外部ホストに戻すことができる。

【００８１】タスク２１６においては、ＣＣ１１はプログラムを適切な処理エンジンに関連するシャドウメモリからインストールする。上で述べたように、データユニットは該データユニットが処理される用意ができるまで処理エンジンに関連するシャドウメモリに留っている。タスク２１６はまたそのチャネルプログラムに対するコンテクストの処理エンジンへのインストールに関与する。

【００８２】いくつかのデータユニットに対して、特定のデータユニットに対する処理エンジンはＦＳ－ＲＩＳＣ２６（図１）である。この状況では、アプリケーションプログラムは一般にすでに動作しておりかつ従って、タスク２１６のプログラムインストールのステップは行われる必要がないかもしれない。この状況では、タスク２１６はＣＣ１１がＫＭＣＥ１２に対しデータユニットがメモリ２５のようなＦＳ－ＲＩＳＣ２６に関連するメモリ（例えば、ＦＳ－ＲＩＳＣ２６に対するメイルボックス）へとロードされておりかつ処理の用意ができていることを通知するタスクを含む。

【００８３】タスク２１８においては、データユニットに関連するコマンドが読み取られる。好ましくは、データユニットのコマンドＤワード６７（図３）は適切な処理エンジンによって読み出され（タスク２１０）、とりわけ、データユニットに関して行われるべき機能を決定する。該処理エンジンは今やデータユニットを処理する用意ができている。処理エンジンがＰＣＥ１４またはＣＣＥ１６である場合、処理エンジンはコマンドＤワード６７をＣＰ－ＲＡＭ９におけるその記憶位置から読み出す。処理エンジンがＦＳ－ＲＩＳＣ２６である場合、ＣＣ１１はコマンドＤワード６７をメモリ２５内のデータユニットのロケーションから読み出す。

【００８４】タスク２１８が行われた後、タスク２２０

はデータユニットを処理する。タスク２１８においてコマンドＩＤフィールド８１を読み出すと、ＣＣ１１は処理エンジンに適切なチャネルプログラムによって該データユニットに対して行われるべき機能を選択させる。タスク２２０においては、選択された機能に関連するキー（単数または複数）およびチャネルが処理エンジンにロードされる。一般に、選択された機能はまた処理されたデータがどこに送られるかを決定する。例えば、暗号機能は処理された（暗号化された）データをＣＴＩＰ１５に送ることができ、一方暗号解読機能は処理された（解読された）データをＰＴＩＰ１３に送ることができる。内部データユニット処理に対しては、処理されたデータはさらに処理するためにＣＰ－ＲＡＭ９に送られ、あるいはＦＳ－ＲＩＳＣ２６によるさらなる処理のためにメモリ２５に送ることができる。

【００８５】典型的な処理機能２２１は暗号化、暗号解読、デジタル署名および真正証明を含む。暗号化に関連しない機能を含む他の機能も行うことができ、キーを使用しない機能を含む。タスク２２２の間に、出力インタフェースプロセッサは処理されたデータユニットを累積する（ａｃｃｕｍｕｌａｔｅｓ）。好ましくは、データユニットの各々のＤワードが処理されると、処理されたＤワードは出力インタフェースプロセッサへと提供される。いったんデータユニットのすべての処理されたＤワードが出力インタフェースプロセッサによって累積されると、出力インタフェースプロセッサはデータユニットが処理を完了したこと、およびインタフェースプロセッサがデータユニットの完全な処理されたペイロード部分を有することを通知される。タスク２２２はまたＡＰＤＵフォーマットのために処理されたデータユニットをフォーマットするタスクを含むことができ、かつ、チャネルヘッダＤワード６６のような、ヘッダ情報およびコマンドＤワード６７（図３）のようなコマンド情報を加えるタスクを含むことができる。タスク２２２はまたデータユニットが処理を完了しており、かつ適切なフォーマットになっていることをＣＣ１１が出力インタフェースプロセッサに通知するステップを含むことができる。

【００８６】タスク２２４においては、インタフェースプロセッサは外部ホストにデータユニットが外部ホストへの転送のために用意ができていることを通知する。好ましくは、外部ホストは該外部ホストが処理されたデータユニットを受ける用意ができたときにデータユニットを要求する。例えば、外部ホストおよび出力インタフェースプロセッサは処理されたデータユニットを転送するためにハンドシェイクプロトコルに関与する（ｅｎｇａｇｅ）ことができる。タスク２２４の一部として、出力インタフェースプロセッサは処理されたデータユニットが転送された後にそのメモリをクリアする。

【００８７】いくつかの場合、データユニットに対して付加的な処理が行われる。タスク２２２において、もし

データユニットに対してさらなる処理が要求されれば、処理されたデータユニットはＰＣＥ１４またはＣＣＥ１６からＣＣ１１へと導き戻される。ＣＣ１１は付加的な処理を予定しかつタスク２１０～２２２が反復される。

【００８８】アプリケーションプログラムはデータユニットに対して付加的な処理がいつ行われるべきかまたは行われるべきことを決定することができる。行われるべき付加的な処理を有するデータユニットはＡＰＤＵとしてフォーマットされＣＣ１１がどの処理を次に予定するかを決定できるようにする。データユニットの処理の実行のシーケンスは好ましくはチャネルプログラムによって決定されかつＣＣ１１によって前記ＡＰＤＵと共にチャネル番号を読み取ることによって実施され実行すべき次のタスクを決定する。

【００８９】本発明の１実施形態では、ＡＰＤＵフォーマットでのデータユニットはタスク２２４の前に再フォーマットされかつ再構成される。例えば、ＡＰＤＵは標準的なＰＤＵフォーマットに変換することができる。この再フォーマット、再構成または変換は出力インタフェースプロセッサによってあるいは外部ホストによって行うことができる。

【００９０】従って、プログラム可能な暗号処理システムが説明され該システムは知られた技術に対して大きな利点を有する。とりわけ、本発明のプログラム可能暗号システムは暗号化、暗号解読およびメッセージの真正証明、メッセージ署名その他のような他の保安サービスなどに対する機能のための大幅に改善された性能を提供する。本発明の処理システムはまた高いグレードの、保安通信システムのための増大する要求に答えることができる。本発明の処理システムはプログラム可能でありかつ単一のＵＬＳＩ設計を使用して複数のプログラムをサポートし、かつ現在のおよび将来の通信装置との共通動作を可能にする。

【００９１】本発明のプログラム可能暗号処理システムおよび方法は複数の暗号プログラムを同時に処理するのに適している。本発明のプログラム可能暗号処理システムおよび方法は処理される各々のデータユニットに関してコンテクストおよびプログラム（例えば、アルゴリズム）の迅速かつ安全なスイッチングを可能にする。

【００９２】とりわけ、本発明のプログラム可能暗号システムは広い範囲のアプリケーションをサポートできる。各々のアプリケーションはいくつかの異なるかつ独立の通信チャネルをもつことができる。さらに、各々のチャネルは異なる暗号変数および状態をもつことができる。本発明のプログラム可能暗号システムのアーキテクチャは正しいプログラムおよび機能がフェイルセイフ動作で実行できることを保証する。

【００９３】本発明のプログラム可能暗号システムはまたフレーミング（ｆｒａｍｉｎｇ）および帯域内シグナリング（ｉｎ－ｂａｎｄ　ｓｉｇｎａｌｉｎｇ）のよう

な非暗号処理をサポートする。好ましい実施形態では、本プログラム可能暗号システムはプログラム可能でありかつシステムが種々の機器種別において使用できるようにし、結果として柔軟性を加えコストをより低下させる。

【００９４】好ましい実施形態では、暗号機能に対する典型的な処理スループットは、種々のプログラムの同時的実行に対するものを含めて、５０ＭＢＰＳのオーダにある。この能力はとりわけ、スループットの利点を提供しならびに複数チャネルの実施をサポートする次のタスクのバックグランドの準備または設定（ｂａｃｋｇｒｏｕｎｄ　ｓｔａｇｉｎｇ）によって達成される。

【００９５】本発明は同時に動作するためにマルチ処理を可能にするアーキテクチャを有するシステムを提供する。例えば、高速コードブック暗号化アプリケーションは高速直列暗号解読ならびにデジタル署名のようなパブリックキー動作と共に同時に動作することができる。本発明のプログラム可能暗号システムは保安手持型無線機から保安高性能マルチチャネル無線機およびネットワークにおよぶ種々のアプリケーションに対して確実な解決方法を提供することを助けるよう区分される。

【００９６】好ましい実施形態では、本発明の処理システムは同時処理能力を備えたスーパースカラーアーキテクチャによってサポートする。高いクロックレートの実行、バケット化されたデータユニットの処理、インテリジェントインタフェースプロセッサおよびバックグランドタスクのスケジューリングによる単一サイクルのタスク交換のための深い（Ｄｅｅｐ）パイプラインマシンが提供される。これらの特徴的機能を１つのシステムへと統合することは標準的な商業的な同様のプロセッサを使用した実施に対してほぼ１０，０００倍の性能上の利点を与える。

【００９７】ＤＳＰの標準的な商業プロセッサによる複数プログラムの処理に伴う問題の１つは保安オペレーティングシステムにおけるタスク交換またはタスクスワッピングによる大きな性能の損失があることである。本発明はバックグランドにおいて高速度の保安タスクスワッピングを可能にする。典型的なマイクロプロセッサおよびＤＳＰに伴う他の問題は高速の暗号化処理に対して高速並列数値および論理処理資源が不十分であることである。本発明のシステムは、好ましい実施形態では、暗号処理を加速する３つの高速処理資源、およびより低い速度の要求に対する低速プロセッサを有する。

【００９８】典型的な暗号処理システムにおいては、キー変数データはサブシステムの間で交換されたとき傷つきやすい（ｖｕｌｎｅｒａｂｌｅ）。この保安性のリスクは本発明によってキー管理およびコントローラサブシステムならびに単一のモノリシックダイ上に配置された暗号処理エンジンによって大幅に低減される。動作機器における深いサブミクロンレベルからのデータの抽出は非常に困難でありかつ従って、最も巧妙な攻撃者からでもデータの喪失を防止することを助ける。本発明のＵＬＳＩ実施形態では、該ＵＬＳＩは好ましくはそのような敏感なデータの保護をさらに強化するためダイの表面のプローピングを防止する保護コーティングによって覆うことができる。

【００９９】暗号プログラム処理は好ましくはハードウェアの障害があった場合でも敏感なデータの喪失を防止することを助けるためフェイルセーフアーキテクチャで行なわれる。典型的には、フェイルセーフ設計は複雑さの増大、コストの増大、電力消費の増大、およびより低い信頼性を生じる結果となる。しかしながら、本発明は好適にフェイルセーフ技術を導入し、これは結果として増大した信頼性、電力消費の低減およびより低いコストを生じることになる。

【０１００】好ましい実施形態では、本発明はまたエンドユーザ機器におけるプログラムのアップグレードを可能にする。暗号プログラムのこの継続的なアップグレードの可能性は保安機器の有用な寿命を延長しかつ他の機器との共用を可能にする。

【０１０１】本発明の好ましい実施形態では、暗号プログラムはそれらが暗号エンジンにロードされるまで暗号保護される。いったん暗号キーまたは暗号化キーが除去されると、プログラムは回復不能にされる。好ましい実施形態では、プログラムソフトウェアを解読するために使用されるキーはゼロ化される（ｚｅｒｏｅｄ）。

【０１０２】したがって、示されたものはプログラム可能暗号処理システムであって、該暗号処理システムはデータユニットを処理するためのプログラム可能暗号プロセッサ（ＰＣＰ）、および各々のデータユニットに含まれる情報に基づき各々のデータユニットに対しチャネルプログラムを識別するための暗号コントローラまたは暗号制御部（ＣＣ）を具備することを特徴とし、かつ前記データユニットの各々はヘッダフィールド、コマンドフィールドおよびペイロード部分からなり、かつ前記ＣＣは前記データユニットの１つのヘッダフィールドを読み取るための手段、前記ヘッダフィールドにおけるチャネルインデクスに基づき前記１つのデータユニットを処理するために複数のチャネルプログラムからチャネルプログラムを識別する手段、前記チャネルインデクスに応じてＰＣＰにおける処理エンジンに対しチャネルプログラムがダウンロードされるようにする手段、そして前記処理エンジンによる処理を予期して前記ペイロード部分をＰＣＰへと転送する手段を具備し、かつ前記コマンドフィールドは処理エンジンによって前記１つのデータユニットに対して行なわれるべき機能を識別し、前記ＰＣＰはさらに、前記１つのチャネルプログラムを記憶するための第１のメモリ、処理エンジンによる前記ペイロード部分の処理に先立ち前記ペイロード部分を記憶するための第２のメモリ、前記機能を決定するために前記１つの

データユニットのコマンドフィールドを読み取るための手段、および前記機能の実行のために前記チャネルプログラムを処理エンジンにロードする手段を具備する。

【０１０３】さらに、システムが示され、該システムにおいてはデータユニットはヘッダフィールド、コマンドフィールドおよびペイロード部分からなり、かつ前記ＰＣＰは前記ペイロード部分を記憶するための第１のメモリおよび複数のチャネルプログラムを記憶するための第２のメモリを含み、前記チャネルプログラムの１つは前記データユニットの前の処理の間に前記データユニットの１つの処理を予期して処理エンジンの第２のメモリにダウンロードされる。

【０１０４】さらに、外部ホストからデータユニットを受信しかつ処理されたデータユニットを外部ホストに転送する複数のインタフェースプロセッサ（ＩＰ）を有するシステムが示され、この場合ＰＣＰはデータユニットの１つの処理された部分を処理エンジンによる該１つのデータユニットの処理の間にインタフェースプロセッサの第２のものに転送する手段を有し、かつ前記第２のインタフェースプロセッサは前記１つのデータユニットがＰＣＰによる処理を完了したことを外部ホストに通知する手段を有し、かつ前記インタフェースプロセッサの第１のものは外部ホストからデータユニットを非同期的に受信する手段を含み、かつ前記第２のインタフェースプロセッサは処理されたデータユニットを外部ホストに非同期的に転送する手段を有する。

【０１０５】また、ＰＣＰ、ＣＣおよび第１および第２のインタフェースプロセッサが単一のダイ上に製造されるシステムが示されている。

【０１０６】さらにシステムが示され、この場合前記データユニットはヘッダフィールド、コマンドフィールドおよびペイロード部分からなり、ＰＣＰはデータユニットに対して機能を達成するため少なくとも２つの処理エンジンを具備し、かつＣＣは、１つのデータユニットのヘッダフィールドを読み取るための手段、該ヘッダフィールドにおけるチャネルインデクスに基づき複数のチャネルプログラムからチャネルプログラムを識別する手段、該チャネルプログラムに基づき処理エンジンの１つを選択する手段、チャネルインデクスに応じてチャネルプログラムがＰＣＰにおける選択された処理エンジンにダウンロードされるようにする手段、そして選択された処理エンジンによる処理を予見して前記ペイロード部分をＰＣＰに転送する手段を具備する。

【０１０７】さらに、ＣＣと結合されたキー管理暗号エンジン（Ｋｅｙ　Ｍａｎａｇｅｍｅｎｔ　Ｃｒｙｐｔｏ　Ｅｎｇｉｎｅ：ＫＭＣＥ）によって特徴付けられるシステムが示され、かつこの場合ＰＣＰはさらにプログラム可能暗号エンジン（ＰＣＥ）および構成可能な暗号エンジン（ＣＣＥ）を具備し、かつＣＣは各々のデータユニットに含まれるチャネルインデクスに基づき各々のデータユニットを処理するための暗号エンジンの１つを選択するための手段、およびチャネルインデクスに応じて各々のデータユニットを暗号エンジンの選択された１つに導く手段を具備し、前記選択された暗号エンジンは各データユニットに対して複数のチャネルプログラムの１つを実行し、かつ前記１つのチャネルプログラムは関連するコンテクストを有し、該コンテクストは外部メモリに暗号化形式で記憶され、かつ前記システムは前記コンテクストを暗号解読して前記コンテクストが１つのデータユニットの処理の前に選択された暗号エンジンに転送されるようにする手段を有する。

【０１０８】さらに、ＣＣが各々のデータユニットに対して複数のチャネルプログラムから１つのチャネルプログラムを識別する手段を有するシステムが示され、かつこの場合ＣＣは各チャネルプログラムに関連するコンテクストを識別する手段を有し、かつＰＣＰは識別されたチャネルプログラムおよび関連するコンテクストにしたがってデータユニットの各々を処理するための手段を有し、前記関連するコンテクストはチャネルプログラムに対する状態（ｓｔａｔｅ）情報およびキーによって特徴付けられる。

【０１０９】また、ヘッダ部分、コマンド部分および関連するペイロード部分を有するデータユニットを処理するためのシステムが示され、この場合前記ヘッダ部分は関連するデータユニットを処理するためのチャネルプログラムを識別し、かつ前記コマンド部分は関連するデータユニットのペイロード部分に対して実行されるべき機能を識別し、前記システムは、前記データユニットの各々によって特定されるチャネルプログラムにしたがって前記データユニットの各々を処理するためのプログラム可能暗号プロセッサ（ＰＣＰ）、および前記ヘッダ部分を読み取りかつ前記関連するデータユニットによって識別されるチャネルプログラムがＰＣＰにおける処理エンジンにダウンロードされるようにする暗号制御部または暗号コントローラ（ＣＣ）を具備することを特徴とし、前記ＣＣは前記ペイロード部分がチャネルプログラムによる処理を待機するため前記処理エンジンのメモリに転送されるようにする。

【０１１０】さらに、システムが示され、該システムは外部ホストからデータユニットを非同期的に受信しかつ該データユニットの１つが処理のための利用できることをＣＣに通知する第１のインタフェースプロセッサ、およびＰＣＰから前記１つのデータユニットの処理された部分を受信し、外部ホストに前記データユニットの処理された部分の全てがＰＣＰから受信されたことを通知し、かつ処理されたデータユニットを外部ホストに非同期的に転送するための第２のインタフェースプロセッサを有する。

【０１１１】さらに、キー管理暗号エンジン（ＫＭＣＥ）を特徴とするシステムが示され、かつこの場合前記

処理エンジンはプログラム可能暗号エンジン（ＰＣＥ）であり、かつＰＣＰはさらに構成可能な暗号エンジン（ＣＣＥ）を具備し、かつＣＣはデータユニットによって識別されたチャネルインデクスに基づき１つのデータユニットを処理するための暗号エンジンの１つを選択する手段、および前記１つのデータユニットをチャネルインデクスに応じて選択された暗号エンジンに導く手段を具備し、前記選択された暗号エンジンは前記１つのデータユニットに対してチャネルプログラムを実行し、かつ前記暗号エンジン、前記ＣＣおよび前記第１および第２のインタフェースプロセッサは単一のダイ上に製造され、前記チャネルプログラムは関連するコンテクストを有し、該コンテクストは外部メモリに暗号化された形式で記憶され、かつ前記システムはコンテクストを暗号解読し該コンテクストが１つのデータユニットを処理する前に選択された暗号エンジンに転送されるようにする手段を有する。

【０１１２】さらに、複数の処理エンジンを有する処理システムにおけるデータユニットを処理する方法が示され、該方法は、データユニットの第１のものにおける情報に基づき複数のチャネルプログラムから１つのチャネルプログラムを識別する段階、前記第１のデータユニットを処理するために前記複数の処理エンジンから１つの処理エンジンを識別する段階、前記第１のデータユニットを前記識別された処理エンジンに関連するメモリに導く段階、前記識別されたチャネルプログラムを識別された処理エンジンにロードする段階、そして前記識別されたチャネルプログラムを使用して前記識別された処理エンジンにおいて前記第１のデータユニットを処理する段階を具備することを特徴とし、かつ前記チャネルを識別する段階はさらに前記チャネルに関連するコンテクストを識別する段階を含み、前記コンテクストはメモリに記憶され、かつ前記処理を行なう段階は前記識別されたチャネルプログラムによる第１のデータユニットの処理の段階を含み、前記識別されたチャネルプログラムは関連するコンテクストを使用する。

【０１１３】さらに１つの方法が示され、該方法はチャネルプログラムを識別する段階、処理エンジンを識別する段階、第１のデータユニットを処理する段階の実行の間に第２のデータユニットのためにルーティングおよびロードを行なう段階を反復する段階を具備することを特徴とし、前記反復されたチャネルプログラムを識別する段階は第２のデータユニットのための第２のチャネルプログラムを識別し、かつ前記識別された処理エンジンは第１の処理エンジンであり、かつ前記方法はさらに、データユニットの第３のものを処理するために第２の処理エンジンを識別する段階、前記第３のデータユニットのための第３のチャネルプログラムを識別する段階、前記第３のデータユニットを前記第２の処理エンジンに関連するメモリに導く段階、前記第３のチャネルプログラムを第２の処理エンジンにロードする段階、そして第２の処理エンジンにおいて前記第３のデータユニットを前記第３のチャネルプログラムを使用して処理する段階を具備することを特徴とし、前記第３のデータユニットを処理する段階は前記第１の処理エンジンによる前記第１のデータユニットを処理する段階と同時的に行なわれる。

【０１１４】さらに１つの方法が示され、この場合前記データユニットはヘッダフィールド、コマンドフィールドおよびペイロード部分からなり、該方法はさらにデータユニットのヘッダフィールドを読み取る段階、前記ヘッダフィールドにおけるチャネルインデクスに基づき前記データユニットを処理するために複数のチャネルプログラムからチャネルプログラムを識別する段階、前記チャネルプログラムに基づき処理エンジンを選択する段階、前記チャネルインデクスに応じてチャネルプログラムを選択された処理エンジンにダウンロードする段階、前記処理エンジンによる処理を予期して前記処理エンジンに関連するメモリロケーションにペイロード部分を転送する段階、外部ホストから非同期的にデータユニットを受信する段階、および処理されたデータユニットを外部ホストに非同期的に転送する段階を具備することを特徴とする。

【０１１５】さらに、複数の処理ユニットを有するプログラム可能暗号処理システムにおける暗号機能を同時に行なう方法が示され、該方法は、第１のヘッダフィールド、コマンドＩＤフィールドおよびペイロード部分からなる第１のデータユニットを受信する段階、第１のヘッダフィールドに基づき第１のデータユニットに対して暗号機能の１つを行なうために処理ユニットの１つを選択する段階、第１のデータユニットを選択された１つの処理ユニットに導く段階、そして選択された１つの処理ユニットがコマンドＩＤフィールドにおける情報に基づきペイロード部分に対して暗号機能の選択された１つを実行する段階を具備することを特徴とする。

【０１１６】また、前記実行段階を行なう間にインタフェースプロセッサにおいて第１の処理されたデータユニットを形成する方法が示され、この場合該方法は前記第１の処理されたデータユニットが形成されたとき外部ホストに通知する段階を含む。また、１つの方法が示され、該方法においては前記ルーティング段階は前記第１のデータユニットを選択された１つの処理ユニットに関連するメモリに導く段階を含む。

【０１１７】さらに、１つの方法が示され、該方法は、前記第１のデータユニットに対して前記実行段階を行なっている間に第２のデータユニットに関して、暗号機能の１つを選択する段階、処理ユニットの１つを選択する段階、および前記導く段階を含んでいる。

【０１１８】また、１つの方法が示され、この場合前記１つの処理ユニットを選択する段階はさらに前記暗号機能の１つを達成するため前記複数の処理ユニットの内の

利用可能な１つを選択する段階を具備する。

【０１１９】さらに、１つの方法が示され、該方法においては前記１つの暗号機能を達成する段階は前記１つの暗号機能に関連するキーをロードする段階、および暗号機能を達成するために該キーを使用する段階を含む。

【０１２０】さらに１つの方法が示され、該方法はさらに前記暗号機能の１つを選択する段階、前記処理ユニットの１つを選択する段階、前記導く段階、および第２のデータユニットのために暗号機能の選択された１つを実行する段階を反復する段階を具備することを特徴とし、前記第２のデータユニットは一連の受信されたデータユニットにおける第１のデータユニットに続くデータユニットであり、前記第１および第２のデータユニットは外部ホストから非同期的に受信される。

【０１２１】また、１つの方法が示され、該方法はさらに複数のチャネルプログラムを否定する段階を具備することを特徴とし、各々のチャネルプログラムは暗号機能およびキーに関連している。

【０１２２】１つの方法が示され、該方法はさらに複数のチャネルを再規定または再定義する段階を具備することを特徴とし、各々のチャネルは暗号機能および暗号キーの組合わせと関連し、各々のデータユニットのヘッダフィールドは複数のチャネルプログラムの１つを識別し、かつ前記実行する段階は各々のデータユニットについて１つのチャネルプログラムに対し暗号キーによって暗号機能の１つを実行する段階を含み、かつ前記暗号機能は暗号化機能を具備し、かつ前記受信段階はシステムの平文（ｐｌａｉｎ－ｔｅｘｔ）プログラム可能インタフェースによって平文で第１のデータユニットを受信する段階を含み、前記選択段階は前記チャネルプログラムに関連する暗号機能の１つを選択する段階を含み、かつ前記１つの暗号機能を実行する段階はシステムに記憶されかつチャネルプログラムと関連する暗号化キーを使用して第１のデータユニットの少なくともペイロード部分を暗号化する段階を含み、かつ前記暗号機能は暗号解読機能を具備し、かつ前記受信段階はシステムの暗号文（ｃｉｐｈｅｒ－ｔｅｘｔ）プログラム可能インタフェースによって暗号文で前記第１のデータユニットを受信する段階を含み、前記選択する段階はチャネルプログラムに関連する１つの暗号機能を選択する段階を含み、かつ前記暗号機能を実行する段階はシステムに記憶されかつチャネルプログラムに関連する暗号キーおよび選択された暗号機能を使用して第１のデータユニットの少なくともペイロード部分を暗号解読するステップを含む。

【０１２３】さらに１つの方法が示され、この場合暗号機能はデジタル署名機能を含み、かつ前記受信段階はシステムのプログラム可能インタフェースにおいて第１のデータユニットを受信する段階を含み、前記選択段階はチャネルプログラムに関連する暗号機能を選択する段階を含み、かつ暗号機能を行なう段階は選択された暗号機能およびチャネルプログラムに関連するシステムに記憶された暗号キーを使用して少なくとも前記第１のデータユニットをデジタル的に署名する段階を含む。

【０１２４】さらに１つの方法が示され、この場合前記暗号機能は真正証明機能を含み、かつ前記受信段階はシステムのプログラム可能インタフェースにおいて第１のデータユニットを受信する段階を含み、前記選択する段階はチャネルプログラムに関連する暗号機能を選択する段階を含み、かつ前記暗号機能を実行する段階は選択された暗号機能およびチャネルに関連するシステムにおいて記憶された暗号キーを使用して前記第１のデータユニットを真正証明する段階を含む。

【０１２５】さらにまた１つの方法が示され、この場合前記ヘッダフィールドは第１のデータユニットに関連するデータユニットの保安レベルを識別するフィールドを含み、かつ前記第１のデータユニットは第１のチャネルプログラムを識別し、第１のチャネルプログラムは関連するプログラム保安レベルを有し、かつ前記方法は前記データユニットの保安レベルをプログラム保安レベルと比較する段階を含み、かつ前記暗号機能を実行する段階は前記プログラム保安レベルが少なくとも前記データユニット保安レベルと同じ大きさである場合に行なわれる。

【０１２６】さらにデータユニットを処理する方法が示され、該方法は、第１のデータユニットから第１のチャネル情報を読み取る段階、第１のチャネル情報によって識別される第１のチャネルプログラムにしたがって第１のデータユニットを処理する段階、第２のデータユニットから第２のチャネル情報を読み取る段階、第２のチャネル情報によって識別される第２のチャネルプログラムにしたがって第２のデータユニットを処理する段階、第１のチャネル情報を読み取る段階に応じて第１のチャネルプログラムを処理エンジンにダウンロードする段階、そして前記第２のチャネル情報を読み取る段階に応答して前記第２のチャネルプログラムを処理エンジンにダウンロードする段階を具備することを特徴とし、前記第２のチャネルプログラムをダウンロードする段階は第１のデータユニットを処理する段階の実行の間に行なわれる。

【０１２７】また、さらに他の方法が示され、該方法はさらに第２のチャネルプログラムに関連するコンテクストを処理エンジンに関連するメモリにロードする段階を具備することを特徴とし、該コンテクストをロードする段階は第１のデータユニットを処理する段階の実行の間に行なわれる。

【０１２８】さらに１つの方法が示され、該方法では処理エンジンは暗号処理システムの複数の処理エンジンの１つであり、前記方法はさらに第１のデータユニットに含まれる情報に基づき前記処理エンジンの１つを識別する段階、および前記第１のデータユニットを前記処理エ

35

ンジンの識別された１つに導く段階を具備することを特徴とし、前記第１のデータユニットを処理する段階は前記処理エンジンの識別された１つによって第１のデータユニットを処理する段階を具備し、かつ前記第１のチャネルプログラムをダウンロードする段階は第１のチャネルプログラムを識別された１つの処理エンジンにダウンロードする段階を具備する。

【０１２９】さらに１つの方法が示され、この場合前記読み取り段階、処理段階、ダウンロード段階およびロード段階はプログラム可能暗号処理システムによって行なわれ、前記コンテクストはシステムの外部のメモリロケーションに記憶され、かつ前記方法はさらに前記コンテクストをロードする段階の前に前記コンテクストを暗号解読する段階、第３のデータユニットから第３のチャネル情報を読み取る段階、該第３のデータユニットに含まれる情報に基づき処理エンジンの内の第２のものを識別する段階、第３のデータユニットを前記処理エンジンの内の第２のものに導く段階、および前記第３のチャネル情報によって識別される第３のチャネルプログラムにしたがって第２の処理エンジンにおいて第３のデータユニットを処理する段階を具備する。

【０１３０】さらに１つの方法が示され、この場合前記第３のチャネル情報を読み取る段階、処理エンジンの第２のものを識別する段階、および第３のデータユニットを導く段階は、前記第１のデータユニットを処理する段階と同時的に行なわれる。

【０１３１】以上の特定の実施形態に関する説明は本発明の包括的な性質を完全に開示しており、これによって他のものが、現在の知識を適用することにより、包括的な概念から離れることなくそのような特定の実施形態を容易に変更しおよび／または種々の用途に適用させることができ、したがってそのような適用および変更は開示された実施形態と同等の意味および範囲内に含まれるものと考えられるべきである。

【０１３２】ここで使用された語法または用語法は説明のためのものであり制限的なものでないことが理解されるべきである。したがって、本発明は添付の特許請求の範囲の精神および広い範囲内に含まれる全てのそのような置換え、変更、等価物および変形を含むことを意図している。

36

＊【図面の簡単な説明】

【図１】本発明の好ましい実施形態に係わるプログラム可能暗号処理システムを示すハードウェアブロック図である。

【図２】本発明の好ましい実施形態に係わるデータユニットの処理を示す説明図である。

【図３】本発明の好ましい実施形態と共に使用するのに適したデータユニットを示すフォーマット図である。

【図４】本発明の好ましい実施形態において使用するのに適したチャネルヘッダを示すフォーマット図である。

【図５】本発明の好ましい実施形態において使用するのに適したコマンドＤワードを示すフォーマット図である。

【図６】本発明の好ましい実施形態において使用するのに適したチャネル識別テーブルを示す説明図である。

【図７】本発明の好ましい実施形態において使用するのに適したプログラムアドレステーブルの一例を示す説明図である。

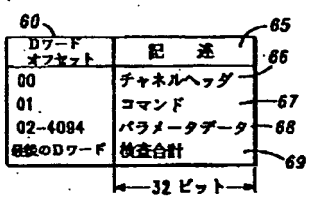
【図８】本発明の好ましい実施形態において使用するのに適したセットアップおよび構成手順を示すフローチャートである。

【図９】本発明の好ましい実施形態において使用するのに適したデータユニットの処理手順を示すフローチャートである。

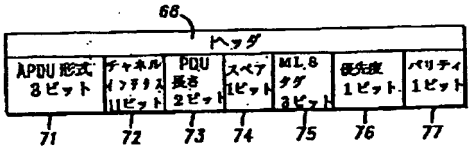
【符号の説明】

１０　暗号処理システム
１２　キーマネージメント暗号エンジン（ＫＭＣＥ）
１３　平文インタフェースプロセッサ（ＰＴＩＰ）
１４　プログラム可能暗号エンジン（ＰＣＥ）
１６　構成可能暗号エンジン（ＣＣＥ）
１７　プログラム可能暗号プロセッサ（ＰＣＰ）
１８　共用メモリ
１９　暗号制御部
２０　テストインタフェース
２１　プログラム可能インタフェース
２２　ｍｏｄＮ解抽出器（ＮＳＥ）
２４　暗号プロセッサＲＡＭ
２５　内部メモリ
２６　フェイルセーフ縮小命令セットコンピュータ（ＦＳ－ＲＩＳＣ）

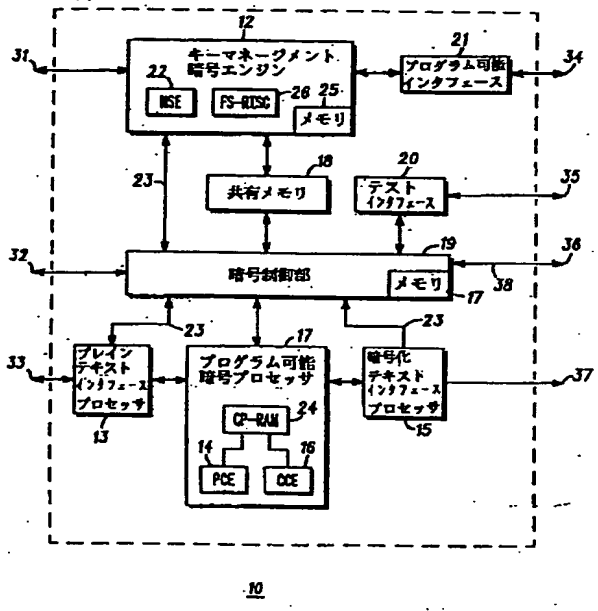
【図３】



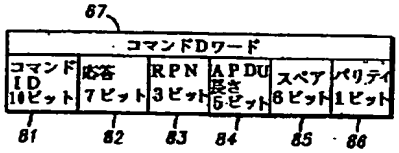
【図４】



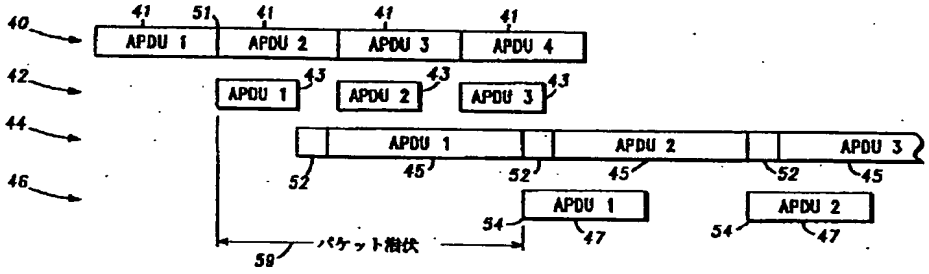
【図1】


12
キーマネージメント暗号エンジン
21
プログラム可能インタフェース
31
34
22
MSE
26
FS-RISC
25
メモリ
18
共有メモリ
20
テストインタフェース
23
35
32
暗号制御部
19
メモリ
36
38
17
23
23
33
プレインテキストインタフェースプロセッサ
13
プログラム可能暗号プロセッサ
CP-RAM
24
14
16
PCE
OCE
暗号化テキストインタフェースプロセッサ
15
37
10

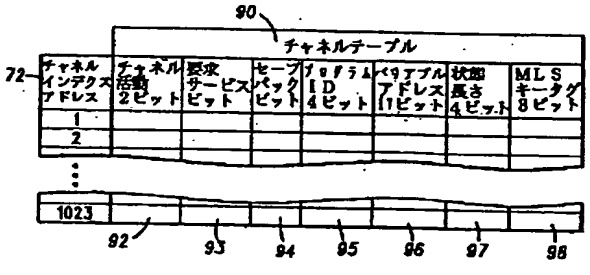
【図5】


67
コマンドDワード
コマンドID 10ビット
応答 7ビット
RPN 3ビット
APDU長さ 6ビット
スペア 6ビット
パリティ 1ビット
81
82
83
84
85
86

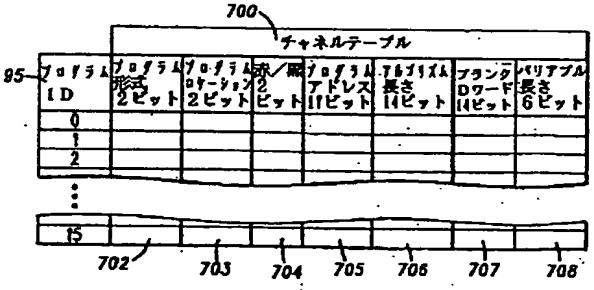
【図2】


40
41
51
APDU 1
APDU 2
APDU 3
APDU 4
42
43
44
45
52
46
54
47
58
パケット樹状

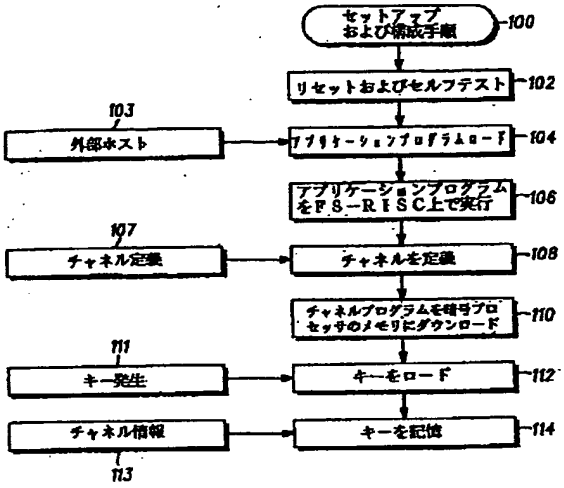
【図6】


90
チャネルテーブル
72
チャネルインデクスアドレス
チャネル活動 2ビット
要求サービスビット
ピープバックビット
プログラムID 4ビット
パリアブルアドレス 11ビット
状態長さ 4ビット
MLSキータグ 8ビット
1
2
1023
92
93
94
95
96
97
98

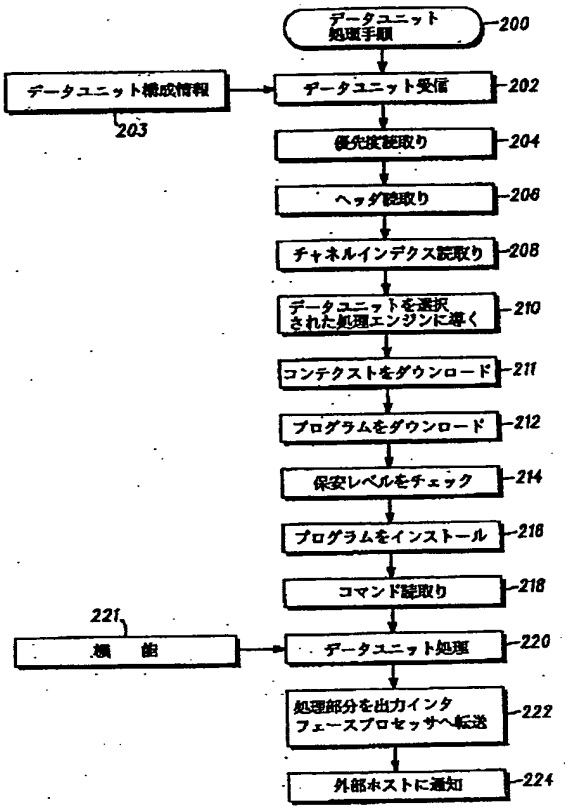
【図7】


700
チャネルテーブル
95
プログラムID
プログラム形式 2ビット
プログラムロケーション2 2ビット
R/W ビット
プログラムアドレス 11ビット
アルゴリズム長さ 14ビット
ブランクDワード 14ビット
パリアブル長さ 6ビット
0
1
2
15
702
703
704
705
706
707
708

【図８】



【図９】



フロントページの続き

(72)発明者　ジェームズ・エドワード・グリーンウッド・ジュニア
アメリカ合衆国アリゾナ州85250、スコッツデイル、レッドウィング・ドライブ
8414

(72)発明者　ケリー・ルシル・ジョンズーバノ
アメリカ合衆国アリゾナ州85254、スコッツデイル、イースト・マリリン・ロード
6328

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-320191
(43)Date of publication of application : 04.12.1998

(51)Int.Cl. G06F 9/06
G09C 1/00
G09C 1/00

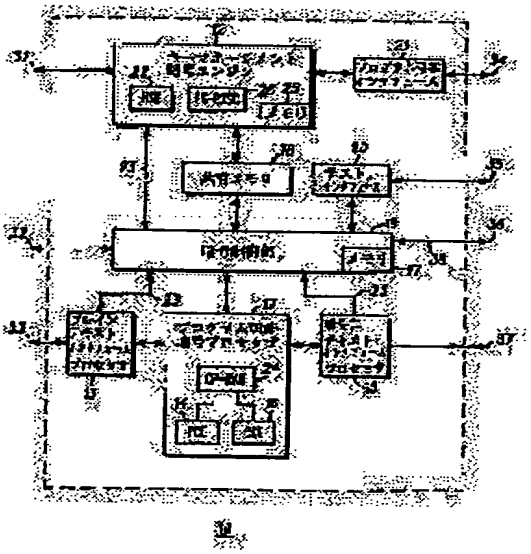
(21)Application number : 10-132755
(22)Date of filing : 27.04.1998

(71)Applicant : MOTOROLA INC
(72)Inventor : HARRISON DAVID MICHAEL
GREENWOOD JAMES EDWARD JR
JOHNS-VANO KERRY LUCILLE

(30)Priority
Priority number : 97 841314 Priority date : 30.04.1997 Priority country : US

(54) PROGRAMMABLE CIPHER PROCESSING SYSTEM AND ITS METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To attain an improved programmable cipher processing system including some processing resources to be executed on a single ULSI die.
SOLUTION: The programmable cipher processing system 10 can quickly correspond to both of a key and algorithm and simultaneously execute various cipher programs by the background staging and cotext (a key and a state) of a succeeding program during the execution of a current program. The system 10 includes a programmable cipher processor 17 for processing a data unit based on a channel program, a cipher control part 19 for identifying the channel program and two interface processors 13, 15 for asynchronously transferring/ receiving data units to/from an external host. Each data unit identifies a specific channel program and is processed by a processing engine selected based on the identified channel program.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] The programmable code processing system which is a processor possible code processing system (10), and is characterized by providing the code control unit (CC) for identifying the channel program to each data unit based on the information included in the programmable code processor (PCP) (17) and each data unit for processing a data unit, and (11) (10).
[Claim 2] Each of said data unit contains the header field, a command field, and a payload part. Said CC The means for reading the one header field of said data unit, The means for discriminating said channel program from two or more channel programs, in order to process said one data unit based on the channel index in said header field, The means for making it said channel program download in the processing engine in said PCP according to said channel index, The means for expecting processing with said processing engine and transmitting said payload part to said PCP, Provide and said command field identifies the function which should be performed to said one data unit with said processing engine. And the 1st memory for said PCP to memorize said one channel program further, The 2nd memory for memorizing said payload part in advance of processing of said payload part with said processing engine, The means for reading said command field of said one data unit, in order to opt for said function, And the means for loading said channel program to said processing engine for activation of said function, Provide and said data unit contains the header field, a command field, and a payload part. And said PCP contains the 2nd memory for memorizing the 1st memory and two or more channel programs for memorizing said payload part. One of said the channel programs is a programmable code processing system characterized by expecting one processing of said data unit between processings before said data unit, and downloading in said 2nd memory of a processing engine.
[Claim 3] It is a data unit processing system for processing the data unit which has a part for a header unit, a command part, and a related payload part. The amount of said header unit identifies the channel program for processing said related data unit. Said command part identifies the function which should be performed to the payload part of said related data unit. And said system The programmable code processor for processing each of said data unit according to said channel program specified by each of said data unit (PCP) (17), And it is the code control section (CC) which makes the processing engine in said PCP download said channel program which reads a part for said header unit, and is identified by said related data unit, and (11). This CC is a data unit processing system characterized by providing what transmitted to the memory of said processing engine in order that said payload part may stand by processing by said channel program.
[Claim 4] It is the approach (200) of processing a data unit in the processing system which has two or more processing engines. The phase of discriminating a certain channel program from two or more channel programs based on the information in the 1st thing of said data unit, (208) The phase of discriminating a certain processing engine from said two or more processing engines in order to process said 1st data unit, (208) The phase of leading said 1st data unit to the memory relevant to said identified processing engine, (210) The phase which loads said identified channel program to said identified processing engine, (216) And (220) the phase of processing said 1st data unit in said identified processing engine using said identified channel

program, Provide and the phase of identifying said channel includes the phase of identifying the context relevant to said channel further. This context is memorized by memory and said phase to process includes the phase of processing said 1st data unit by said identified channel program. Said identified channel program is the approach (200) of processing a data unit in the processing system which has two or more processing engines characterized by using said related context.

[Claim 5] It is the approach (200) of performing the code function in the programmable code processing system which has two or more processing units instantaneous. The phase of receiving the 1st data unit containing the 1st header field, command ID field, and payload part, (202) The phase which chooses one of said the processing units in order to perform one of said the code functions to said 1st data unit based on said 1st header field, (208) The phase of leading said 1st data unit to said one selected processing unit, (210) The phase where said one selected processing unit performs one as which said code function was chosen to said payload part based on the information in said command ID field (220), And the phase which forms the data unit by which the 1st was processed in the interface processor between achievement of said phase to perform, Provide and said approach includes the phase which notifies an external host of said data unit by which the 1st was processed having been formed. And said phase to draw is the approach (200) of performing the code function in the programmable code processing system which has two or more processing units characterized by including the phase of leading said 1st data unit to the memory relevant to said one selected processing unit instantaneous.

[Translation done.]

* NOTICES *

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] Generally this invention relates to the field of security cryptocommunication.
[0002]
[Description of the Prior Art] The inclination in a communication link commercial scene has specified the need for security nature (security) to the both sides of the object for commerce, and the market for military affairs clearly. It is important to respond to equip communication system with complicated communication service and capacity, and become elaborateness more, to preserve and to keep information safe. One of the problems accompanying a security device is protection of the code program from exploitation by the reverse engineering technique. Generally it is thought that the operation by the hardware of the code program by which the code program is included in hardware is safe. The problem accompanying operation of hardware is that that to which an adversary or an interest is opposed can determine a program by die inquiry (die probing) and analysis using extraordinary efforts. Other problems of the code system which is carried out or consists of hardware are semi-conductor processings of the high cost for the chip which processes a code program. A semi-conductor is manufactured under a security condition and it is because the code program is included in hardware logic.
[0003] Carrying out the deer of the code program which is carried out by software or is constituted, it is thought that the configuration of hardware is not more typically safe and it is for the accessibility of software. The typical problem accompanying the configuration of software is that simultaneous processing of two or more programs produces loss of the engine performance by the task exchange (task swapping) in a security operating system as a result. Other problems accompanying the configuration of software are that the arithmetic and logic unit of a typical microprocessor and a digital signal processor does not have juxtaposition, the numeric value, and logic process resources of a high speed desirable for high-speed cipher processing.
[0004]
[Problem(s) to be Solved by the Invention] the time of the problem accompanying the code processing system of both hardware and software being exchanged between subsystems — the defenselessness of a key variable data — or it is attacked and is easy (vulnerability). This is the risk of the general security nature for today's code system.
[0005] Therefore, that it is the need is the code processing system and approach which have been improved. It is the code processing system and approach of being processed in a commercial chip fabrication factory, excluding a code program (crypto programs), and reducing the cost of semi-conductor processing that it is furthermore the need. Moreover, the code system for the code program manipulation of high performance is also required. Furthermore, the code system which can perform two or more programs is also required for coincidence. It is a code processing system quick (keyand algorithm agile) to a key and an algorithm, and an approach that it is furthermore the need. It is quick, the code processing system which can change the context (context) and program (for example, algorithm) over each data unit processed by insurance, and an approach that it is furthermore the need. Furthermore, when exchanged between different subsystems, code SHISUMUTE which protects a key variable data

is required. That it is furthermore the need is a code system by which a code program is protected from reverse engineering.
[0006]
[Means for Solving the Problem] This invention is divided and offers a programmable code processing system and a programmable approach. This invention offers the code processing system suitable for processing the code program of high performance again. This invention provides coincidence with the system and approach of processing two or more code programs again. This invention offers the code processing system and approach of changing the context and program (for example, algorithm) over each data unit processed again to a high speed and insurance. This invention offers the code processing system and approach of protecting a key variable data, when further exchanged between different subsystems. This invention offers the system and approach of having been suitable for processing the code program in the architecture of a failsafe again. This invention offers the programmable code processing system which reduces the semi-conductor processing cost relevant to a still more typical code processing system. With a desirable operation gestalt, the security nature of a key variable data is protected, when exchanged between subsystems. Moreover, in a desirable operation gestalt, a code program can be updated in the device arranged in the site. Moreover, with a desirable operation gestalt, a code program is protected from reverse engineering.
[0007]
[Embodiment of the Invention] This invention is especially pointed out to the attached claim. However, a more perfect understanding of this invention can be acquired by referring to the following detailed explanation and claims with an attached drawing. In a drawing, the same reference figure has mentioned the same item over a drawing.
[0008] Drawing 1 shows the hardware block diagram of the programmable code processing system (crypto processing system) concerning the desirable operation gestalt of this invention. The code processing system 10 has the two main processing elements, the key management code engine (Key management crypto engine:KMCE) 12, and the programmable code processor (programmable cryptographic processor:PCP) 17 with a desirable operation gestalt. PCP17 possesses two processing engines, the programmable code engine (programmable cryptographic engine:PCE) 14, and the code engine (configurable cryptographic engine:CCE) 16 that can be constituted. Said processing engine performs a channel program. In a system 10, this CC11 performs program management for a processing engine again including the code controller (cryptographic controller:cc) 11. A system 10 contains the plain text or the plaintext interface processor (plane text interface processor:PTIP) 13, and the cipher interface processor (cipher text interface processor:CTIP) 15 which offer signaling for an external interface and a system 10 again. Said interface processor offers the high performance security flexible buffer between an external host and the internal-processing system of a system 10 again. A system 10 contains the common use or the shared memory 18 which acts as a resiliency **** buffer between KMCE12 and PCP17 again. A system 10 includes the programmable interface 21 combined with FILL and the CIK port 34 again. The trial of a system 10 can be performed using the trial interface 20 including an emulation on chip and the JTAG port 35.
[0009] KMCE12 is combined with CC11 by the internal bus 23, including an internal memory 25. Other internal buses 23 combine PTIP13, CTIP15, PCP17, and a shared memory 18 with CC11.
[0010] With a desirable operation gestalt, KMCE12 contains the reduced instruction-set computer (FS-RISC) 26 of a failsafe again. KMCE12 contains the 2nd desirable process resources like MODDO or the modulo N solution extractor (mod N solution extractor:NSE) 22. FS-RISC26 consists of a 32 (dual) bit RISC core of a duplex preferably, and this performs the security (embeded) operating system (secure operating system:SOS) incorporated or embedded. This security operating system offers segmentation (segmentation) and task management, in order that a task may enable it to perform from the program memory of the exterior of a system 10. Or such a task does not perform security processing, it can contain the task and subroutine which do not deal with delicate data (sensitive data). The task and subroutine which perform security processing or deal with delicate data are performed from the internal program memory (ROM) preferably contained in memory 25.
[0011] With the desirable operation gestalt of this invention, by SOS of said FS-RISC, the

function performed from Interior ROM is divided and includes master control of a system 10, the self-test of a system 10 and an alarm monitor, a program load, and real-time multi-level security task management. A program load loads the both sides of security and a non-preserving program to an internal memory 25, or includes loading to an algorithm or PCP17 of a program.

[0012] FS-RISC26 can operate the application software from the internal program memory (RAM) of memory 25 again. The typical application software which operates by FS-RISC26 from the internal program RAM includes a function like the fill port processing for CIK and low processing of delicate data, or port restoration processing (fill-port processing). This example includes loading of a key. The example of other application software which operates includes generating, other key managements, and the control function of a session key for example, by the public key program. Application software can include loading (loading), verification (verifying), modification (changing), a system management like auditing (auditing), and a key function manager again.

[0013] FS-RISC26 can operate the application software from external program memory again. Such external program memory can be set to RAM of an external host system. Such application software that operates from external program RAM includes a function like the software irrelevant to processing of interface protocol processing (for example, DS-101 and NSA 87-27), key management operation, command processing, non-preserving program software, and delicate data directly preferably.

[0014] PCP17 is a high performance programmable superscalar (superscaler) cipher-processing element which divides, and performs the function about a data unit, and processes a data unit. a data unit — desirable — an external host — an interface processor 13 — or it is loaded to an interface processor 15. CC11 starts processing of a data unit by acting as Ince Tan Scheidt (loading) of the context (context) demanded, a program code, a condition (state), and the variable according to the read of the header information of a data unit. If a data unit is once loaded to PCP17 and processing is performed, a result will be written in an output interface processor. the processed data unit — or in order to process further, it can provide for other destinations like KMCE12.

[0015] CC11 divides and manages synthetic data migration between the activation resources of interface processors 13 and 15 and the code engines 14 and 16, NSE22, and FS-RISC26. CC11 is insurance by the data which should move roughly, the task which should be installed in PCP17, and determining when program execution is started — it operates as a ****** real-time operating system again. CC11 attains this by investigating the contents of each data unit. This is later explained to a detail. This data drive architecture provides a system 10 with the throughput of high performance. Furthermore, CC11 performs a background dead work or a background staging (background staging). A following task and a following data unit are set up or prepared between activation of a current task (staged). Said background activity makes the high throughput for a system 10 possible. For example, the program load for a transfer of the data unit to PCP17, the clean-up of memory, and the following data unit is performed between processings of a front data unit.

[0016] With the desirable operation gestalt of this invention, PCP17 divides and possesses two high-speed processing engines which perform other data processing typically performed in a function like channel encryption and decryption, a security communication link, and signaling, and PCE14 and CCE16. With a desirable operation gestalt, PCE14 programs a code book format (codebook style), and, on the other hand, CCE16 performs the program (combiner style) of a combiner format. PCE14 and CCE16 operate independently, and are combined, and offer a bigger throughput than 1200MIPs to 32-bit data. PCE14 and CCE16 are constituted from the 32 bit RISC processor of high performance which operates by about 100MHz with 4 stage pipeline configuration by the desirable operation gestalt of this invention. These RISC processors can be divided and can be used also for data processing like other protocols specified by signal processing, error detection, correction, and the channel program in a band (inch-band), and format processing.

[0017] PCP17 contains the code processor RAM 9 (CP-RAM) for memorizing a channel program and/or a data unit again. CC11 downloads a channel program from CP-RAM9 to the

memory of a processing engine, before processing a data unit. CC11 downloads the context of a channel program from CP-RAM9 to the memory of a processing engine, before processing a data unit again.
[0018] KMCE12 divides and attains the master control function for a system 10. With a desirable operation gestalt, KMCE12 contains the security operating system (SOS) built into ROM in KMCE12. With a desirable operation gestalt, FS-RISC26 is a 32 bit RISC processor of high performance. In addition to FS-RISC26, KMCE12 contains the math coprocessor or mass co-processor (math coprocessor) which was preferably suitable for processing of a public key program. With this operation gestalt, KMCE12 has the throughput of about 150 MIPs, in order to enable activation of the application (embeded) with which the multiple channel and the single channel were embedded.
[0019] With other operation gestalten, a system 10 can act as a cipher-processing element embedded for various applications. For example, a system 10 enables it to carry out data flow through architecture (data flow through architectures) or co-processor architecture (coprocessor architecture). In data flow through architecture, it can pass through data cipher interface port 37 from the plaintext interface port 33, or they can flow to the reverse. It helps for the internal security mechanism incorporated or embedded in the system 10 to isolate or separate delicate (sensitive) plaintext data and a logically different data type like a variable from the cipher data protected. With the configuration of co-processor architecture, in order that a host system may isolate said type or the data of a format, for example, the certainty (design assurance) of a big design is offered suitably.
[0020] With the desirable operation gestalt of a system 10, PTIP13 and CTIP15 are equipped with a FIFO control structure, and include 8 bits, 16 bits, and 32 bit-parallel data interface in ports 33 and 37. Interface processors 13 and 15 also include serial asynchronous ones and a serial synchronous interface preferably. PTIP13 and CTIP15 include an internal processor, an internal physical memory, and external memory escape capacity. The memory of an interface processor is managed by those internal processors. With a desirable operation gestalt, full-duplex (full duplex) actuation is possible for an interface processor, and in order to process a plaintext and cipher data, it offers the isolation of a perfect physical data interface.
[0021] The interface port 31 is connected with KMCE12, and includes the port for a memory interface, a configuration signal (configuration signals), a system clock, and interruption preferably. A memory interface port is constituted from the control interface for accessing a 33 bits data bus, a 24-bit address bus and an internal memory, or an I/O device by the desirable operation gestalt. With the desirable operation gestalt of a system 10, KMCE12 receives a command and data through PTIP13 or CTIP15. As for other operation gestalten, control and data can come from the interface port 31.
[0022] A system 10 contains the context memory bus 38 (CNTX) connected to the context port (context port) 36 again. The context memory bus 38 is constituted from the 33-bit data bus used in order to combine with external context memory, and an address control bus by the desirable operation gestalt. CC11 manages exchange or the swapping of the context from the task in PCP17 to the active inactive task in external context memory. A bus 38 enables context change of the high speed for the application which requires many instantaneous tasks rather than what may exist in an internal memory. A port 32 offers the interface to CC11 for a control signal and the alarm signal according to individual.
[0023] The context (Context) currently used here can include the functional information relevant to a condition (state) or variable information (variable information), a key, and a channel, including the information relevant to a specific channel program.
[0024] With a desirable operation gestalt, the code processing system of this invention is preferably carried out on a single silicon die in very-large-scale-integrated-circuit (ULSI) equipment. With a desirable operation gestalt, some processing subsystems are accumulated into said ULSI, and the throughput of about 1350 suitable MIPs can be acquired for the class of a wide range code program.
[0025] Drawing 2 shows processing of the data unit concerning the desirable operation gestalt of this invention. The architecture of the code processing system of this invention enables processing of the packet-ized communication link thread (threads) of the multiple channel

equipped with the very high throughput. The asynchronous operation between an internal subsystem and an external host is managed by the finite-state machine (finite state machine) in CC11.

[0026] Reference of drawing 2 transmits the data unit 41 to one of the interface processors 13 or 15 (drawing 1) from an external host, as shown by a time amount line or the time line (time-line) 40. An interface processor is notified by sending that it is ready to CC11 for processing of the new data unit 41 to time amount 51, and sending the header of this data unit to CC11. Based on the information on the header of the data unit 41, it is ordered for CC11 to move a data unit to a suitable processing subsystem like KMCE12, PCE14, or CCE16 at an interface processor. Preferably, all for example, other than a header are transmitted to a processing engine for a part of data unit.

[0027] When it is prepared so that a data unit may be processed by one of the engines in PCP17, CC11 plans processing and begins. By mediation, CC11 makes a data unit transfer max in order to make simultaneous processing in a system 10 max preferably. In the time line 42, the data unit 41 is transmitted to memory like CP-RAM9 (drawing 1), and it stands by in order to be processed by the suitable processing engine (for example, PCE14 or CCE16) there. This background activity of the following data unit which should be processed makes the incubation (latency) covering a system 10 min. Furthermore, it helps to guarantee that the resource of PCE14 or CCE16 is processing the data unit, and the background activity of a program is not moving data or a program. Therefore, the data throughput of a system increases sharply.

[0028] The time line 44 shows the period which has transmitted a part for the data division which the processing engine was processing the data unit 45, and was processed to the output interface processor. A time frame 52 is one clock cycle typically, and is context switching time to which a key and a program are changed between them. What the interface processor was ready to process a new data unit to CC11 in time amount 51 is notified. The data unit processed between time line 44 is transmitted to an output interface processor from a processing unit. Processing of a data unit is completed by time amount 54. The data unit has completed processing to an external host, and an output interface processor notifies an available thing to this time amount. The data unit 47 is a processed data unit, and is transmitted to an external host between time line 46. The incubation 59 of the packet relevant to processing this data unit is shown as time amount to the time amount which is ready to transmit the data unit processed from reception of the packet in an input interface processor to an external host.

[0029] After the whole packet is received by the processing unit (PCP17), a data unit is processed by the processing engine (for example, PCE14 or CCE16) so that it may see from the processing diagram of drawing 2 . Furthermore, a data unit cannot be used in order to transmit to an external host until the whole data unit is processed. It is transmitted with the continuous (continual) base as a data unit consists of two or more desirable D WORD (Dwords) (32-bit WORD), and the each is processed separately and then processing is performed from a processing unit to an output interface processor. With a desirable operation gestalt, after the whole data unit completes processing and an external host becomes available in an output interface processor, he is notified. Processing of a perfect data unit makes the work which avoids the deadlock or deadlock which may be produced by the action or action from an external host.

[0030] An output interface processor is an interface processor relevant to the interface port of the opposite side where a data unit is generated or sent from there typically. For example, after it is processed, the data which generate in a plain text or the plaintext interface port 33 are sent to CTIP15, and are made available in the cipher interface port 37.

[0031] With a desirable operation gestalt, a data unit is loaded to interface processors 13 or 15 in asynchronous by the host system, and is managed by the interface processor. The data unit planned or planned for activation by PCE14 or CCE16 is sent to the memory relevant to a processing unit (for example, CP-RAM9), and is memorized. When a processing engine is FS-RISC26, the data unit which was ready for processing is memorized by memory 25. Interface processors 13 and 15 perform a function like a decomposition [of a data unit] (dataunit parsing), priority attachment (prioritizing), and juxtaposition-serial and serial-parallel conversion,

packet integration, inspection or check word (checkword) generating, and a memory management function.
[0032] With the desirable operation gestalt of this invention, the data unit processed by the system 10 is formatted specially because of processing by the system 10. In this operation gestalt, interface processors 13 and 15 process data in the APDU format explained below. However, the stream data (stream data) which are not in an APDU format are also received in juxtaposition or the serial port of an interface processor, and it can change into an APDU format for processing.
[0033] Drawing 3 shows a format of the data unit suitable for using it with the desirable operation gestalt of this invention. The data unit in an APDU format is shown in drawing 3 . The data unit of an APDU format consists of a series of D WORD. Each D WORD has the offset shown in a column 60. The first D WORD is D WORD 66 of a channel header, and this is 32-bit D WORD preferably. There is command D WORD 67 following D WORD 66 of a channel header, and this has D WORD offset of 1. There is a parameter data field 68 which has D WORD offset between 2 and 4094 following command D WORD 67. The parameter data field 68 of APDU contains the payload (application payload) of application. It is because the data of the field 68 can have a different format to each channel and an application program can carry out context exchange (context swap) of it to each data unit. For example, probably, some channels need the strong (robust) protocol in order to guarantee the lock step processing to a communication link thread (communication thread), while two or more programs are performing in a system 10 to two or more channels.
[0034] The last D WORD is a checksum or checksum (check sum) D WORD 69, and this is a 32-bit frame check sequence (frame check sequence:FCS) preferably calculated over the whole APDU. A checksum or (Checksum CS) D WORD 69 is the option-field which can enable or carry out a disable to a specific application. It judges whether between start-up procedures, KMCE12 constituted CC11 and the checksum was attached to each APDU.
[0035] One suitable FCS program is the 32-bit version of an ISO3309-1964E specification. This specification specifies the high-level data link control procedure and the frame structure for information processing system and data communication.
[0036] Drawing 4 shows a format of the channel header suitable for using it in the desirable operation gestalt of this invention. A format of this channel header specifies the size or the magnitude and location, or location of the field in D WORD 66 of a channel header. Channel header D WORD 66 contains the MLS tag field 75 of the PDU die-length field 73 of 72 or 12 bits of channel index fields of 71 or 19 bits of APDU type fields of a triplet, the spare bit 74, and a triplet, the priority bit 76, and a parity bit 77. Option-like [the MLS tag field 75 and the priority bit 76]. The APDU type field 71 specifies the value over an APDU type and its corresponding meaning, or significance (significance). Preferably, the APDU type field 71 specifies the source of APDU from PTIP13, CTIP15, or other internal sources of a system 10. The APDU type field 71 also shows the output processor which should receive a data unit preferably.
[0037] It specifies whether the APDU type fields 71 are whether APDU is Demand APDU again and Response APDU. To Response APDU, the channel index field 72 already contains the demand program number (request program number:RPN) of the triplet instead given in the command D WORD of Demand APDU excluding a channel index. Using the APDU type field 71, CC11 is divided and opts for use (use) of the channel index field 72.
[0038] a ****** [that the channel index field 72 is calling the channel usual in a data unit] — or it specifies whether the data unit is calling the internal resource. For example, if the bit of the beginning of a channel index is "1", 10 bits of the last will identify the channel program used in the channel table explained later. A channel table specifies the property of a channel. In case a context and a program are moved to the active channel memory of an EU and CC11 is taken out from active channel memory, it manages a channel table. When a channel is generated, an entry is added to this channel table. the time of the entry of a channel table being removed — the channel — being inactive (inactive) — it becomes. The table of an inactive channel is moved to the storage location which a condition (state) and a variable data, and/or a program cannot access by the condition machine of C11. The application program which operates on

FS-RISC26 can carry out rediscount reliance of the channel program from this table, and can remove data from PCP17. The memory used in order to memorize inactive channel data can be prepared in KMCE12 or external context memory.
[0039] About the channel index field 72, if the bit of the beginning of a channel index is zero, as for a data unit, the internal resource may be demanded for processing. 10 bits next to this channel index show which internal resource is demanded. An internal resource contains Randa Myser (randomizer) in PTIP13, CTIP15, CC11, and PI21, and FS-RISC26.
[0040] The PDU die-length field 73 shows the number of D WORD containing the option CS tooth-space D WORD which follows command D WORD 67 preferably. The die-length field 73 specifies the size or magnitude of an application data. With the operation gestalt shown in drawing 3 , the size of the greatest application-data unit is 4094D WORD, and this is 131,008 bits.
[0041] The MLS tag field 75 specifies the levels of security of APDU. The value in the MLS tag field 75 is compared with the value of the MLS tag of the key relevant to a channel by the desirable operation gestalt. When two tags do not have consistency, a data unit is eliminated and an error situation is set. With the desirable operation gestalt of this invention, the MLS tag of a key is loaded with this key, or is specified at the time of key creation. The MLS tag of this key is based on the levels of security used in order to create this key preferably.
[0042] The priority bit 76 specifies the priority level for APDU. This priority bit is preferably used by interface processors 13 or 15, and chooses the sequence of processing of a data unit. There is a priority of two level with the shown operation gestalt. For example, zero specify non-real-time (non-real-time) processing, and on the other hand, since real-time processing is specified, "1" is used.
[0043] A parity bit 77 is preferably added to each header D WORD. CC11 is inspected when this header is processed in the parity of this header WORD.
[0044] Drawing 5 shows a format of the command D WORD suitable for using it in the desirable operation gestalt of this invention. Command D WORD 67 is 2nd D WORD in each APDU preferably. Command D WORD 67 contains the response field 82 of 81 or 7 bits of 10 bits of command ID fields, the APDU die-length field 84 of 83 or 5 bits of demand program-number (RPN) fields of a triplet, the spare bit 85, and a parity bit 86. With a desirable operation gestalt, command ID field 81 specifies the function which should be performed to a data unit. A function is preferably specified to each channel program. There is [as opposed to / at a desirable operation gestalt / a system 10] no original function. A function can contain encryption, decryption, a sign, the Shinsei certification, and others. For example, a function like encryption specifies what a part for the data division of APDU (for example, parameter data field 68) should be enciphered for to application software. This encryption is performed using the key specified to the channel chosen by a channel program and the channel index field.
[0045] A response field 82 returns the processing status with the processed data unit. This response is generated by the processing unit of a system 10. For example, PCE14 can provide a response field 82 with the "completion of processing (processing complete)" response in the end of transmission of the data unit to an output interface processor. Similarly, CC11 can send a "default" response value to a transmitting processor, when a transfer of a data unit goes wrong. It shall depend for a response field 82 on specific application or a specific channel program.
[0046] The RPN field 83 is used in order to identify which program published the demand in APDU (request type) of a demand format. Since it maps in one of the processings which is operating by one of for example, the encryption engines now, CC11 can use RPN. The RPN field 83 returns the value in the response APDU channel index by which CC11 enables it to lead APDU to a right processor. With a desirable operation gestalt, when APDU generates from an external host, the RPN field is not used and is set to zero. By identifying a program, the RPN field 83 requires and delivers a command, a parameter, and data between different channel programs which are operating in the EU of a system 10. Since a processing unit can run a program to coincidence, a processing unit uses a data unit with communication link structure again. Therefore, a program can transmit information among these very thing by use of the RPN field 83 using CC11.

[0047] The APDU die-length field 84 specifies the size of APDU. A parity bit 86 is added to the header of command D WORD 67. CC11 can inspect the parity about this header WORD, when it processes command D WORD 67 (drawing 3).
[0048] Drawing 6 shows the channel convention or channel definition table suitable for using it in the desirable operation gestalt of this invention. The channel index field 72 (drawing 4) of header D WORD 66 (drawing 4) determines the line (row) of the channel table 90 which is read by CC11 (drawing 1) and applied to APDU. A channel table 90 specifies the contents and those die length of the channel table field. With a desirable operation gestalt, in order to characterize each channel, let a channel table 90 be a 1024 word die-length x32 bit table. CC11 uses the field of a channel table 90, when setting up a channel program in the processing engines 14 and 16. A channel table 90 contains the die-length field 97 of 96 or 4 bits of adjustable address fields of 95 or 17 bits of program ID fields of the assigned 2-bit activity field (allocated activity field) 92, the demand service bit 93, and 94 or 4 bits of save back bits (save back bit), and the MLS key tag field 98 of a triplet.
[0049] The information on a channel table 90 is used in order to lead APDU to suitable process resources, and it includes other information for install of the specific communication link thread to the channel, or re-install. Generally, a channel table contains the pointer to the location by which a program and a context are arranged for a channel definition or a convention. The channel index field 72 can also be directed to the channel which is not assigned to PCP17 (point). In this case, CC11 can lead a data unit to KMCE12 to which processing is performed. Generally, processing is performed by KMCE12 with the exception base (exception basis).
[0050] The channel index field 72 is assigned by the software of the application program performed on FS-RISC26, and is produced at the time of creation/convention of a channel. It is fixed depending on specific application, or let allocation of a channel index be a dynamic thing. For example, an external host enables it to build APDU appropriately by it including exchange of the value [channel assignment / dynamic] at the time of channel creation. When a new channel is created or pulled down (torn down), KMCE12 creates a new entry in a channel table 90, or deletes an entry. The channel table 90 in a desirable operation gestalt is stored in the memory 19 of CC11.
[0051] Each channel has the related channel condition (channel state) stored immanent in CC11. the current program state to which the channel condition is operating, the condition of a degree or the last, a standby condition, and an install condition — and inactive or an inactive condition is included. A channel is in operating state (running state), when a current program state and a current context are performing on PCE14 or CCE16. PCE14 and CCE16 have at least four memory of a lot, and these enable it to load the following channel with a desirable operation gestalt, while it is chosen in a ping-pong (ping-pong) format and the current channel is carrying out current activation. Since this memory swapping occurs, said memory convention or definition changes from it being active (active) to a shadow (shadpw).
[0052] The channel condition of a degree or the last shows that a channel program exists in the shadow memory relevant to PCE14 or CCE16 which were described above. It specifies that a standby channel condition is ready so that an application program may exist in CP-RAM9 and it may be installed in a shadow memory. An install channel condition is in the channel condition between standby and the thing of a degree or the last, when swapped with the thing of others [context / relevant to the channel program to one channel]. An inactive channel condition is in the condition which has said context and/or program out of control of PCP17. for example, a program — KMCE12 — or it can exist in external context memory.
[0053] Once a channel is established in PCP17, an external host's application can process APDU without the mediation from the application program currently performed in FS-RISC26 in PCP17 with the base for every channel. Therefore, the greatest throughput between functions like encryption or decryption is attained by the autonomous (per channel) processing for every channel in PCP17. Therefore, typical application delivers APDU without mediation of KMCE12 through PCP17.
[0054] The channel activity field 92 identifies an effective channel, and includes channel status information. The data unit which identifies an invalid channel program can be written in FS-RISC26 for processing. The channel activity field 92 shows the processing activities of the

channel, when effective. When the data unit is processed, the channel activity field 92 is updated. It follows and determines which channel for the channel activity field 92 to be used by KMCE12 again, and not to be used more rarely and whether to be removable with the minimum effect on a system 10. With a desirable operation gestalt, the channel activity field 92 is updated through a number "01", "10", and "11" one by one. The current or present value is memorized with the channel currently used in order to process specific APDU. The value of the channel activity field 92 expresses the condition of having been used for the last of a channel.
[0055] It is shown that the demand service bit field 93 needs to read a new value to the comprehensive variable (global variables) by which the application program currently performed on PCE14 or CCE16 was updated with the application program currently performed on FS-RISC26. Therefore, when the demand service bit or the service request bit is set, FS-RISC26 offers additional information, before a program begins. The save back bit 94 shows how to use it, since CC11 saves a context. for example, a context — CP-RAM9 — or it is savable to external memory. After a context changes some of contexts or all in which the channel program generally introduced or installed in one of the processing engines is existing, it is returned. By using the save back bit 94, many APDU(s) can operate on the same channel and can produce change in the context. When a channel is removed from one of the processing engines, the save back of the context is carried out to the interior or external memory. Therefore, unnecessary save is avoided.
[0056] Program ID field 95 contains the ID code for an algorithm or a program. Preferably, program ID field 95 directs the line (row) in the program address table which exists in the memory 19 of CC11. A program address table divides and specifies the field used in order to pursue the program from which CC11 differs. A variable or the adjustable address field 96 specifies the starting address of the memory location in PCP17 in which adjustable or the variable data to a channel is located. By using adjustable or the variable address field 96, it is determined whether CC11 is in whether this variable is in current active memory or it is in a shadow memory, a variable, or condition memory. Furthermore, the variable address field 96 has a variable data in CP-RAM9 to CC11, and shows the thing of PCE14 of CCE16 which should be moved to active or a shadow memory to the number condition of browning. a ****** [that the shadow and the memory address to active memory were fixed preferably, therefore activation / CC11 / of APDU is ready] — or it enables it to determine SUTAGU [APDU / in a shadow memory] While SUTAGU [APDU / the shadow memory], a channel parameter to the channel like a program variable and a condition is loaded to a processing engine.
[0057] The condition die-length field 97 specifies the die length of the state-variable data described in the top. With a desirable operation gestalt, the condition die-length field 97 is changed between zero and D WORD of 32. The MLS key tag field 98 enumerates the levels of security of a channel key (lists). The value of the key tag field 98 is compared with the tag received in the MLS tag field 75 of header D WORD 66 of APDU. There should be the levels of security of the channel key enumerated by the key tag field 98 more highly than the data security level identified in the MLS tag field 76 of header D WORD 66 to the data unit which should be processed.
[0058] Drawing 7 shows the example of the program address table suitable for using it in the desirable operation gestalt of this invention. The program address table 700 includes the program type field 702, the program location field 703, the red / black (red/black) field 704, the program address field 705, the program die-length field 706, the blank D WORD field 707, and the variable die-length field 708. Program ID field 95 (drawing 6) from a channel table 90 directs the line (row) of the channel-address table 700. Therefore, each channel is connected with the line in the program address table 700.
[0059] The program type field 702 is the 2-bit field which identifies [of a channel program] being size, for example, is large?, or whether it is small. The program type field 702 identifies the EU, PCE14 or CCE16, with which a program operates again. [for example,] The program location field 703 identifies the location of the channel program to the channel. CC11 opts for the location of a program, when required, in order that it may process APDU using the program location field 703. The program location field 703 shows when a program should be loaded by FS-RISC26 again. The program location field can show that only one copy of the program exists

again, and it exists in a processing engine like PCE14 or CCE16 eternally. the program location field 703 — or what a channel program is in CP-RAM9, and should be copied when required for a suitable processing engine is shown. It is shown that the program location field 703 has a program in external memory again, and is copied to a system 10 if needed. When a program is in external memory, this program may need to be decoded by KMCE12 before install in one of the processing engines.

[0060] Red / black field 704 is the 2-bit fields which identify the levels of security of a program preferably. Red / black field 704 is divided and it is shown whether this program is a security program or it is not a security program. After it is darkness, or it is enciphered before being moved to black (black) external memory, and a security program moves this program from external memory, it should be decoded. Before a non-preserving program moves to external memory or moves from external memory, it does not need to be enciphered. With the desirable operation gestalt of this invention, red / black field 702 is not used, when external memory is not used for a program store or it cannot use because of a program store.

[0061] The program address field 705 contains the address pointer which identifies the memory location of a channel program or the program over the channel. This memory location can be in PCE14, CCE16, CP-RAM9, or external memory. CC11 determines the location of a channel program using the program address field 705, and moves it into the shadow memory of a processing engine. When a specific program is eternally loaded to a processing engine, said program address can include the value which exists in order to show that a program does not need to move.

[0062] The program die-length field 706 identifies the size of the microcode of the channel program memorized by memory. The blank D WORD field 707 shows the number of the zero or blank D WORD which CC11 writes in a memory location, after CC11 installs a program in the memory. Said zero or blank D WORD is written in the program installed before to the program tooth space of a continuation processing engine, in order to guarantee that the exaggerated light of the program tooth space is carried out.

[0063] The variable die-length field 708 contains the die length of the variable used in this specific program. The die length of a variable can be made the same to all the channels that use the same program. The die length of a program variable is between zero and 32D WORD preferably. CC11 uses said variable die length, when installing the context of a channel in a processing engine.

[0064] Drawing 8 is the flow chart of the setup suitable for using it in the desirable operation gestalt of this invention, and a configuration (configuration) procedure. Since it divides, and it specifies and a definition or a related channel program is loaded for a channel to PCP17 by the system 10, a procedure 100 is performed. The programmable code processing system of this invention can have some programs which operate to coincidence by the superscalar (superscaler) programmable architecture. These programs are installed from the master application program which operates on the security operating system of FS-RISC26. In a task 102, KMCE12 performs reset and self-test processing in order to guarantee that the component (components) and subsystem of a system 10 are operating appropriately. In a task 104, a master application program is loaded to KMCE12 from the external host 103. With another operation gestalt of this invention, an application program exists in the memory 25 of KMCE12, and is loaded to FS-RISC26 from memory 25.

[0065] In a task 106, the application program loaded in the task 104 is performed, and it performs on the security operating system of FS-RISC26 preferably.

[0066] In a task 108, it is ordered an application program so that two or more channels may be created and defined to CC11 using the channel definition information 107. Channel definition information or the channel convention information (Channel definition information) 107 is memorized in a system 10, or can be offered by the external host. Between this step, a channel table as shown in a channel table 90 (drawing 6) is created. Furthermore, a program address table as shown in the program address table 700 of drawing 7 is also created. These tables are memorized by a share or the shared memory 18 (drawing 1) with the desirable operation gestalt of this invention. Between the tasks 108 of a setup and configuration procedure 100, a channel program is not preferably installed in the processing engines 14 or 16. A channel

program is installed for a specific data unit, when a data unit is processed. For example, the channel index of APDU runs to CC11 (run), and a channel program sets it to it, CC11 installs this program, and this program execution is started in a suitable processing engine.
[0067] The channel definition information 107 includes the information which defines or specifies the specific program of a context, or relation with a program segment. The single thread (thread) of an activation code (execution code) is the example of a channel. In the multi-processing system for which a context is exchanged, it operates by time sharing for multi-channel actuation of many instantaneous channels. Therefore, the separate context to each channel is maintained preferably.
[0068] After a channel is defined and a channel program is identified in a task 110, an application program downloads a specific channel program in memory like CP-RAM9 of PCP17. In relation to each channel, there is a channel program preferably.
[0069] In a task 112, a code key (encryption keys) is loaded to a system 10. Preferably, this key is loaded to the programmable interface 21 through the fill port (fill port) 34. A key contains a DES code key and the keys of other formats which could set on the technique of public, a private key, and cryptography, and were known including the key used for encryption, decryption, a digital signature, and the Shinsei certification. With a desirable operation gestalt, memory 25 has a backup power supply like a dc-battery in order to prevent loss of a key in the case of the power failure to a system 10. A task 112 can include the key generating task 111 which generates a key in FS-RISC26 in option. It can include that key generating performed by FS-RISC26 is public or using private key generating software. A channel or a session key can be generated by many approaches technically learned including what is depended on FS-RISC26 using internal Randa Myser (randomizer). A key is memorized with a desirable operation gestalt by the table to which it relates to a channel and a channel is related with a suitable key or a key pair. With a desirable operation gestalt, a key is connected with each channel between initialization of FS-RISC26. With 1 operation gestalt, the levels of security of a channel relate it with a specific key.
[0070] If the task 112 includes the key generating task 111, a key shall be used for a key escrow (key escrow). A task 112 can include the task which provides a key escrow with a key again. A key is memorized in a task 114 by memory like [in order to use it when it is connected with the channel which uses the channel information 113 and a data unit is processed] the local memory relevant to CP-RAM9, or PCE14 or CCE16. If a task 114 is completed, the system 10 is ready to process a data unit.
[0071] Drawing 9 is the flow chart of the procedure of the data unit suitable for using it in the desirable operation gestalt of this invention. With a desirable operation gestalt, a procedure 200 is performed to each data unit received by the system 10. Generally, a procedure 200 is made to perform a certain function to each data unit. A function includes encryption, decryption, a signature, or the Shinsei certification. After the data unit which this function was performed and was processed is completed, a system 10 enables it to use the processed data unit for an external host.
[0072] In a task 202, a data unit is received in interface processors 13 or 15 from an external host. The data unit is the APDU format which was preferably explained by drawing 3 - drawing 5 . A data unit is convertible for an APDU format from other formats with an external host. For example, in the case of the stream data (stream data) which are not an APDU format, PTIP13 or STIP15 can format these stream data using the configuration information 203 memorized by the system 10. Although an external host changes data into an APDU format with a desirable operation gestalt before being received by the interface processor in a task 202, what prevents a system from changing [10] a data unit into an APDU format does not have anything.
[0073] Configuration information (Configuration information) 203 includes the specific information based on the application of a system. For example, configuration information 203 can include [the classification of the data unit which should be processed, the interface which should be used, APDU format information, and] when PTIP13 or CTIP15 generate APDU.
[0074] A task 202 can receive a data unit in [it is desirable synchronous and] asynchronous. A data unit is receivable by juxtaposition or the series-connected-type formula through the juxtaposition or the serial port of an interface processor related again. When a data unit is

received in asynchronous, it notifies that an interface processor can be used in order that it may receive a data unit to an external host.
[0075] In a task 204, an input interface processor reads a packet priority (namely, bit 76 of header D WORD 66 (drawing 4)), and the processing to the data unit is planned or planned. With a desirable operation gestalt, the packet equipped with the real-time priority is first sent to CC11, and a non-real-time packet is sent following it. It notifies that the interface processor is ready to process a new data unit to CC11 as a part of task 204. In a task 206, CC11 reads the header of a data unit.
[0076] In a task 208, CC11 determines a suitable channel program and process resources, in order to read the field 71 to an APDU type, and the MLS tag field 75 of header D WORD 66 of a data unit in the field 72 for a channel index and to process this data unit. CC11 can read the APDU die-length field 84 as a part of task 208 again.
[0077] In a task 210, it is ordered CC11 so that a data unit may be led to the processing engine or FS-RISC26 like PCE14 or CCE16 to an interface processor. A processing engine is chosen based on the information from a task 208. The channel index field 72 of header D WORD 66 determines an external unit, in order to process to a data unit. With a desirable operation gestalt, a data unit is led to CP-RAM9, and waits for processing by PSE14 or CCE16 there. Or a data unit is led to the shadow memory of PCE14 or CCE16, and waits for processing by PCE14 or CCE16 there, respectively.
[0078] Between tasks 210, the frame-check-sequence (frame check sequence:FCS) checker in CC11 investigates the integrity of the data unit between transfers. A default response is returned to the external host who offered the data unit when a problem occurs by FCS. The PDU die-length field 73 of header D WORD 66 is used by CC11 in order to allocate memory in PCP17. With 1 operation gestalt of this invention, a task 210 includes leading only the payload part of a data unit to a processing engine.
[0079] The context to the specific channel downloads in a task 211. In a task 212, a channel program downloads CC11 in a suitable processing engine. Preferably, a program is loaded to the shadow memory of PCE14 or CCE16 (drawing 1).
[0080] In a task 214, the MLS tag field 75 is compared with the tag (namely, MLS key tag field 98) in a channel table, and a large thing is guaranteed to levels-of-security extent which the data unit requires [the levels of security of a program] at least. When a data unit requires big security nature rather than a channel provides, this data unit is not processed preferably and a default response is returned to an input interface processor. An interface processor can return this default response to the external host who offered that data unit.
[0081] In a task 216, CC11 installs a program from the shadow memory relevant to a suitable processing engine. As stated in the top, ruble's is in the shadow memory relevant to a processing engine until a data unit is ready to process this data unit. A task 216 participates in install in the processing engine of the context to the channel program again.
[0082] A processing engine [as opposed to a specific data unit to some data units] is FS-RISC26 (drawing 1). In this situation, generally the application program may already have operated, and may follow, and the step of program install of a task 216 may not need to be performed. In this situation, a task 216 includes the task which notifies that CC11 is loaded to the memory (for example, mail box to FS-RISC26) relevant to FS-RISC [like memory 25]26 whose data unit is to KMCE12, and processing is ready.
[0083] The command relevant to a data unit is read in a task 218. Preferably, command D WORD 67 (drawing 3) of a data unit is read with a suitable processing engine (task 210), and is divided, and it opts for the function which should be performed about a data unit. This processing engine is ready to process a data unit now. When a processing engine is PCE14 or CCE16, a processing engine is read from the storage location [in / for command D WORD 67 / CP-RAM9]. When a processing engine is FS-RISC26, CC11 reads command D WORD 67 from the location of the data unit in memory 25.
[0084] After a task 218 is performed, a task 220 processes a data unit. When command ID field 81 is read in a task 218, CC11 makes the function which should be performed to this data unit by the suitable channel program for a processing engine choose. In a task 220, the key (the unit or plurality) and channel relevant to the selected function are loaded to a processing engine.

Generally, the selected function determines where the data processed again are sent. For example, a code function can send the processed data (enciphered) to CTIP15, and, on the other hand, a decryption function can send the processed data (it decoded) to PTIP13. To in-house-data unit processing, the processed data are sent to CP-RAM9, in order to process further, or they can be sent to memory 25 for the further processing by FS-RISC26.

[0085] The typical processing facility 221 includes encryption, decryption, a digital signature, and the Shinsei certification. Other functions including the function irrelevant to encryption can be performed, and the function which does not use a key is included. Between tasks 222, an output interface processor accumulates the processed data unit (accumulates). Preferably, processing of each D WORD of a data unit offers processed D WORD to an output interface processor. Once all processed D WORD of a data unit is accumulated by the output interface processor, it will be notified to an output interface processor that the data unit completed processing and that an interface processor has the processed perfect payload part of a data unit. A task 222 can include the task which can include the task which formats the data unit processed again for the APDU format, and adds header information like channel header D WORD 66, and command information like command D WORD 67 (drawing 3). The step which CC11 notifies that it is that the data unit has completed processing again and the task 222 has become a suitable format to an output interface processor can be included.

[0086] In a task 224, it notifies that the external host of the interface processor is ready for the transfer to an external host of a data unit. Preferably, an external host demands a data unit, when it is ready to receive the data unit by which this external host was processed. For example, an external host and an output interface processor can do what (engage) is participated in a handshake protocol, in order to transmit the processed data unit. As a part of task 224, an output interface processor clears the memory, after the processed data unit is transmitted.

[0087] In some cases, additional processing is performed to a data unit. In a task 222, if the further processing is required from a data unit, the processed data unit will be returned to CC11 from PCE14 or CCE16. CC11 plans additional processing and tasks 210-222 are repeated.

[0088] as for an application program, additional processing should be performed when to a data unit — it is — or what should be performed can be determined. The data unit which has the additional processing which should be performed is formatted as APDU, and CC11 enables it to determine which processing is planned next. The sequence of activation of processing of a data unit determines the following task which should be performed by carrying out by determining a channel program preferably and reading a channel number with said APDU by CC11.

[0089] With 1 operation gestalt of this invention, the data unit in an APDU format is reformatted and reconfigurated in front of a task 224. For example, APDU is convertible for a standard PDU format. this reformatting, reconstruction, or conversion — an output interface processor — or an external host can perform.

[0090] Therefore, a programmable code processing system is explained and this system has a big advantage to the known technique. The programmable code system of this invention especially offers the engine performance improved sharply for the function to encryption, decryption and the Shinsei certification of a message, other security services like a message signature and others, etc. The processing system of this invention can reply to the demand to which it increases for the security communication system of grade high again. Using a programmable and single ULSI design, two or more programs are supported and the processing system of this invention enables common actuation with the communication device of current and the future.

[0091] The programmable code processing system and approach of this invention are suitable for processing two or more code programs to coincidence. The programmable code processing system and approach of this invention enable a context and quick and safe switching of a program (for example, algorithm) about each data unit processed.

[0092] The programmable code system of this invention can especially support the application of the large range. each application — some — differing — and it can have an independent communication channel. Furthermore, each channel can have a different code variable and a

different condition. It guarantees that a right program and a function can perform architecture of the programmable code system of this invention by failsafe operation.
[0093] The programmable code system of this invention supports non-cipher processing like [again] framing (framing) and signaling in a band (inch-band signaling). With a desirable operation gestalt, it enables it to use it in the device classification of versatility [system] with this programmable and programmable code system, flexibility is added as a result, and cost is reduced more.
[0094] With a desirable operation gestalt, the typical processing throughput to a code function is in the order of 50MBPS including the thing to instantaneous activation of various programs. This capacity is attained by the preparation or a setup (background staging) of the background of the following task which offers the advantage of a throughput and especially supports operation of a multiple channel.
[0095] This invention offers the system which has the architecture which enables multi-processing, in order to operate to coincidence. For example, high-speed code book encryption application can operate to coincidence with high-speed serial decryption and public key actuation like a digital signature. The programmable code system of this invention is classified so that it may rescue offering the positive solution approach to the various applications which reach a security high performance multi-channel walkie-talkie and a network from a security stock mold walkie-talkie.
[0096] With a desirable operation gestalt, the processing system of this invention is supported by the superscalar architecture equipped with simultaneous-processing capacity. The deep (Deep) pipeline machine for task exchange of the single cycle by the scheduling of activation of a high clock rate, processing of the packet-ized data unit, an intelligent interface processor, and a background task is offered. Unifying these characteristic functions to one system gives the advantage on one about 10,000 times the engine performance of this to the operation which used the same standard commercial processor.
[0097] One of the problems accompanying processing of two or more programs by the standard commercial processor of DSP is that there is loss of the big engine performance by the task exchange or the task swapping in a security operating system. This invention makes high-speed security task swapping possible in a background. Other problems accompanying a typical microprocessor and typical DSP are that a high-speed juxtaposition numeric value and logic process resources are inadequate to high-speed encryption processing. The system of this invention has three high-speed process resources which accelerate cipher processing, and a low-speed processor to the demand of a lower rate with a desirable operation gestalt.
[0098] In a typical code processing system, a key variable data tends (vulnerable) to get damaged, when exchanged between subsystems. The risk of this security nature is sharply reduced with the cipher-processing engine arranged by this invention on key management, a controller subsystem, and a single monolithic die. It helps for the extract of the data from the deep submicron level in a device of operation to be very difficult, and to follow, and to prevent loss of data even from the most skillful aggressor. With the ULSI operation gestalt of this invention, since this ULSI strengthens protection of such sensitive data further preferably, it can be covered by the protective coating which prevents probing of the front face of a die.
[0099] It is performed by failsafe architecture in order to help to prevent loss of sensitive data, even when code program manipulation has the failure of hardware preferably. Typically, a fail-safe design brings a result which produces increase of complexity, increase of cost, increase of power consumption, and lower dependability. However, this invention will introduce a failsafe technique suitably and this will produce reduction and the lower cost of the dependability which increased as a result, and power consumption.
[0100] With a desirable operation gestalt, this invention enables the upgrade of the program in an end user machine again. The possibility of this continuous upgrade of a code program extends the useful life of a security device, and enables common use with other devices.
[0101] With the desirable operation gestalt of this invention, code protection of the code program is carried out until they are loaded to a code engine. A program will be made into recovery impossible once a code key or a cryptographic key is removed. With a desirable operation gestalt, the key used in order to decode program software is zero-ized (zeroed).

[0102] Therefore, what was shown is a programmable code processing system. A programmable code processor for this code processing system to process a data unit (PCP), And it is characterized by providing the code controller or code control section (CC) for identifying a channel program to each data unit based on the information included in each data unit. And each of said data unit consists of the header field, a command field, and a payload part. And a means for said CC to read the one header field of said data unit, A means to discriminate a channel program from two or more channel programs in order to process said one data unit based on the channel index in said header field, The means which a channel program downloads to the processing engine in PCP according to said channel index, And a means to expect processing with said processing engine and to transmit said payload part to PCP is provided. And said command field identifies the function which should be performed to said one data unit with a processing engine. The 1st memory for said PCP to memorize said one channel program further, The 2nd memory for memorizing said payload part in advance of processing of said payload part with a processing engine, In order to opt for said function, a means to load said channel program to a processing engine for the means for reading the command field of said one data unit and activation of said function is provided.
[0103] A system is shown and it sets to this system. A data unit Furthermore, the header field, Consist of a command field and a payload part, and said PCP contains the 2nd memory for memorizing the 1st memory and two or more channel programs for memorizing said payload part. One of said the channel programs expects one processing of said data unit between processings before said data unit, and it downloads it in the 2nd memory of a processing engine.

[0104] Furthermore, the system which has two or more interface processors (IP) which transmit the data unit which received the data unit from the external host, and was processed to an external host is shown. It has a means to transmit between processings of one data unit at the 2nd thing of an interface processor. in this case, PCP depends one processed part of a data unit on a processing engine — this — And said 2nd interface processor has a means to notify an external host of said one data unit having completed processing by PCP. And said 2nd interface processor has a means to transmit the processed data unit to an external host in asynchronous, including a means by which the 1st thing of said interface processor receives a data unit in asynchronous from an external host.
[0105] Moreover, the system by which PCP, CC, and the 1st and 2nd interface processors are manufactured on a single die is shown.
[0106] Furthermore, a system is shown. Said data unit in this case The header field, It consists of a command field and a payload part, and in order that PCP may attain a function to a data unit, at least two processing engines are provided. And CC The means for reading the header field of one data unit, a means to discriminate a channel program from two or more channel programs based on the channel index in this header field, A means to choose one of the processing engines based on this channel program, According to a channel index, a channel program possesses the means downloaded in the selected processing engine in PCP, and a means to foresee processing with the selected processing engine and to transmit said payload part to PCP.
[0107] Furthermore, the system characterized with the key management code engine (Key Management Crypto Engine:KMCE) combined with CC is shown. And PCP possesses the code engine (CCE) in which a programmable code engine (PCE) and a configuration are still more possible in this case. And the means for choosing one of the code engines for CC processing each data unit based on the channel index contained in each data unit, And according to a channel index, a means to lead each data unit to one as which the code engine was chosen is provided. Said selected code engine performs one of two or more of the channel programs to each data unit. And said one channel program has a related context. This context is memorized by external memory in an encryption format, and said system has the means transmitted to the code engine with which said context was decrypted and said context was chosen before processing of one data unit.
[0108] Furthermore, the system which has a means by which CC discriminates one channel program from two or more channel programs to each data unit is shown. And CC has a means

to identify the context relevant to each channel program, in this case. And PCP has a means for processing each of a data unit according to the identified channel program and a related context, and said related context is characterized by the condition (state) information and the key to a channel program.

[0109] Moreover, the system for processing the data unit which has a part for a header unit, a command part, and a related payload part is shown. In this case, the amount of said header unit identifies the channel program for processing a related data unit. Said command part identifies the function which should be performed to the payload part of a related data unit. And said system The programmable code processor for processing each of said data unit according to the channel program specified by each of said data unit (PCP), And it is characterized by providing the code control section or code controller (CC) which the channel program which reads a part for said header unit, and is identified by said related data unit downloads in the processing engine in PCP. Said CC is transmitted to the memory of said processing engine, in order that said payload part may stand by processing by the channel program.

[0110] Furthermore, the 1st interface processor which notifies to CC that a system is shown, and, as for this system, an external host to a data unit is received in asynchronous, and it can use for one processing of this data unit, And the part in which said one data unit was processed from PCP is received. It has the 2nd interface processor for transmitting in asynchronous the data unit which notified the external host of all the parts by which said data unit was processed having been received from PCP, and was processed to an external host.

[0111] Furthermore, the system characterized by the key management code engine (KMCE) is shown. And said processing engine is a programmable code engine (PCE) in this case. And a means to choose one of the code engines for processing one data unit based on the channel index from which PCP possessed the code engine (CCE) which can further be constituted, and CC was discriminated by the data unit, And a means to lead said one data unit to the code engine chosen according to the channel index is provided. Said selected code engine performs a channel program to said one data unit. And said code engine, said CC, and said 1st and 2nd interface processors are manufactured on a single die. Said channel program has a related context and this context is memorized in the format enciphered by external memory. And said system has the means transmitted to the code engine chosen before it decrypted the context and this context processed one data unit.

[0112] The method of processing the data unit in the processing system which has two or more processing engines is shown. Furthermore, this approach The phase of discriminating one channel program from two or more channel programs based on the information in the 1st thing of a data unit, The phase of discriminating one processing engine from said two or more processing engines in order to process said 1st data unit, The phase of leading said 1st data unit to the memory relevant to said identified processing engine, The phase which loads said identified channel program to the identified processing engine, And it is characterized by providing the phase of processing said 1st data unit in said identified processing engine using said identified channel program. And the phase of identifying said channel includes the phase of identifying the context relevant to said channel further. The phase of said context being memorized by memory and performing said processing uses the context to which said identified channel program relates including the phase of processing of the 1st data unit by said identified channel program.

[0113] The phase where one more approach is shown and this approach identifies a channel program, It is characterized by providing the phase of identifying a processing engine, and the phase which repeats the phase of performing routing and loading between activation of the phase of processing the 1st data unit, for the 2nd data unit. The phase of identifying said repeated channel program identifies the 2nd channel program for the 2nd data unit. Said identified processing engine is the 1st processing engine. Said approach and further The phase of identifying the 2nd processing engine in order to process the 3rd thing of a data unit, The phase of identifying the 3rd channel program for said 3rd data unit, The phase of leading said 3rd data unit to the memory relevant to said 2nd processing engine, The phase which loads said 3rd channel program to the 2nd processing engine, And it is characterized by providing the phase of processing said 3rd data unit in the 2nd processing engine using said 3rd channel

program. The phase of processing said 3rd data unit is performed on the phase and coincidence target which process said 1st data unit with said 1st processing engine.
[0114] One more approach is shown. Said data unit in this case The header field, The phase where consist of a command field and a payload part, and this approach reads the header field of a data unit further, The phase of discriminating a channel program from two or more channel programs in order to process said data unit based on the channel index in said header field, The phase which chooses a processing engine based on said channel program, the phase downloaded in the processing engine which had the channel program chosen according to said channel index, The phase of expecting processing with said processing engine and transmitting a payload part to the memory location relevant to said processing engine, It is characterized by providing the phase of receiving a data unit in asynchronous from an external host, and the phase of transmitting the processed data unit to an external host in asynchronous.
[0115] The method of performing the code function in the programmable code processing system which has two or more processing units to coincidence is shown. Furthermore, this approach The phase of receiving the 1st data unit which consists of the 1st header field, command ID field, and payload part, The phase which chooses one of the processing units in order to perform one of the code functions to the 1st data unit based on the 1st header field, It is characterized by providing the phase led to one processing unit which had the 1st data unit chosen, and the phase where one selected processing unit performs one as which the code function was chosen to the payload part based on the information in a command ID field.
[0116] Moreover, while performing said running phase, the approach of forming the data unit by which the 1st was processed in the interface processor is shown, and this approach includes the phase of which an external host is notified, when said data unit by which the 1st was processed is formed in this case. Moreover, one approach is shown and said routing phase includes the phase led to the memory relevant to one processing unit which had said 1st data unit chosen in this approach.
[0117] Furthermore, one approach is shown, and this approach includes the phase which chooses one of the code functions, the phase which chooses one of the processing units, and said phase to draw about the 2nd data unit, while performing said running phase to said 1st data unit.
[0118] Moreover, one approach is shown, and the phase which chooses said one processing unit in this case possesses the phase which chooses available one out of said two or more processing units in order to attain one of said the code functions further.
[0119] Furthermore, one approach is shown, and the phase of attaining said one code function in this approach includes the phase which uses this key, in order to attain the phase which loads the key relevant to said one code function, and a code function.
[0120] The phase where one more approach is shown and this approach chooses one of said the code functions further, It is characterized by providing the phase which repeats the phase of performing one as which the code function was chosen for the phase which chooses one of said the processing units, said phase to draw, and the 2nd data unit. Said 2nd data unit is a data unit following the 1st data unit in the data unit by which a single string was received, and said 1st and 2nd data units are received in asynchronous from an external host.
[0121] Moreover, one approach is shown, it is characterized by this approach possessing the phase which denies further two or more channel programs, and each channel program relates to the code function and the key.
[0122] One approach is shown and it is characterized by this approach possessing a re-convention or the phase to redefine for further two or more channels. As for the header field of each data unit, each channel identifies one of two or more of the channel programs in relation to the combination of a code function and a code key. And said phase to perform includes the phase of performing one of the code functions by the code key to one channel program about each data unit. And said code function possesses an encryption function, and said receiving phase includes the phase where the plaintext (plain-text) programmable interface of a system receives the 1st data unit by the plaintext. Said selection stage story includes the phase which chooses one of the code functions relevant to said channel program. And the phase of performing said one code function is memorized by the system, and includes the phase of the

1st data unit which enciphers a payload part at least using the cryptographic key relevant to a channel program. And said code function possesses a decryption function, and said receiving phase includes the phase where the cipher (cipher—text) programmable interface of a system receives said 1st data unit by the cipher. Said phase to choose includes the phase which chooses one code function relevant to a channel program. And the phase of performing said code function is memorized by the system, and contains the step of the 1st data unit which decrypts a payload part at least using the code key and the selected code function relevant to a channel program.

[0123] One more approach is shown and a code function includes a digital signature function in this case. And said receiving phase includes the phase of receiving the 1st data unit in the programmable interface of a system. Said selection stage story includes the phase which chooses the code function relevant to a channel program. And the phase of performing a code function includes the phase of signing said 1st data unit in digital one at least using the code key memorized by the system relevant to the selected code function and the selected channel program.

[0124] One more approach is shown and said code function includes the Shinsei certification function in this case. And said receiving phase includes the phase of receiving the 1st data unit in the programmable interface of a system. Said phase to choose includes the phase which chooses the code function relevant to a channel program. And the phase of performing said code function includes the phase which carries out Shinsei certification of said 1st data unit using the code key memorized in the system relevant to the selected code function and the selected channel.

[0125] One approach is shown further again and said header field includes the field which identifies the levels of security of the data unit relevant to the 1st data unit in this case. And said 1st data unit identifies the 1st channel program. The 1st channel program has the related program levels of security. And including a phase [levels of security / of said data unit / the program levels of security / approach / said], the phase of performing said code function is performed, when said program levels of security are the magnitude same at least as said data unit levels of security.

[0126] The method of furthermore processing a data unit is shown. This approach The phase of reading the 1st channel information in the 1st data unit, the phase of processing the 1st data unit according to the 1st channel program identified using the 1st channel information, The phase of reading the 2nd channel information in the 2nd data unit, the phase of processing the 2nd data unit according to the 2nd channel program identified using the 2nd channel information, The phase which downloads the 1st channel program in a processing engine according to the phase of reading the 1st channel information, And it is characterized by providing the phase which answers the phase of reading said 2nd channel information, and downloads said 2nd channel program in a processing engine. The phase which downloads said 2nd channel program is performed between activation of the phase of processing the 1st data unit.

[0127] Furthermore, other approaches are shown, it is characterized by this approach possessing the phase which loads the context relevant to the 2nd channel program to the memory relevant to a processing engine further, and the phase which loads this context is performed between activation of the phase of processing the 1st data unit.

[0128] One more approach is shown and a processing engine is one of two or more of the processing engines of a code processing system by this approach. The phase of identifying one of said the processing engines based on the information by which said approach is further included in the 1st data unit, And it is characterized by providing the phase of leading said 1st data unit to one from which said processing engine was discriminated. The phase of processing said 1st data unit possesses the phase of processing the 1st data unit by one from which said processing engine was discriminated. And the phase which downloads said 1st channel program possesses the phase which downloads the 1st channel program in one identified processing engine.

[0129] One more approach is shown. In this case Said reading phase, a processing phase, A download phase and a load phase are performed by the programmable code processing system.

Said context is memorized by the memory location of the exterior of a system. And the phase of decrypting said context in front of the phase where said approach loads said context further, the phase of reading the 3rd channel information in the 3rd data unit — this — the phase of identifying the 2nd thing of the processing engines based on the information included in the 3rd data unit — The phase of leading the 3rd data unit to the 2nd thing of said processing engines, and the phase of processing the 3rd data unit in the 2nd processing engine according to the 3rd channel program identified using said 3rd channel information are provided.

[0130] The phase of one more approach being shown and reading said 3rd channel information in this case, the phase of identifying the 2nd thing of a processing engine, and the phase of drawing the 3rd data unit are performed on the phase and coincidence target which process said 1st data unit.

[0131] When the comprehensive property of this invention is indicated completely and other things apply current knowledge by this, the explanation about the above specific operation gestalt Such a specific operation gestalt can be changed easily, it can be made to apply to various applications, without separating from a comprehensive concept, therefore such application and modification should be considered to be contained within semantics and limits equivalent to the indicated operation gestalt.

[0132] The usage or nomenclature used here is a thing for explanation, and it should be understood that it is not restrictive. Therefore, it replaces and this invention has the intention of such a thing including modification, an equivalent, and deformation of all that are contained within the pneuma of an attached claim, and large limits.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the hardware block diagram showing the programmable code processing system concerning the desirable operation gestalt of this invention.
[Drawing 2] It is the explanatory view showing processing of the data unit concerning the desirable operation gestalt of this invention.
[Drawing 3] It is the format Fig. showing the data unit suitable for using it with the desirable operation gestalt of this invention.
[Drawing 4] It is the format Fig. showing the channel header suitable for using it in the desirable operation gestalt of this invention.
[Drawing 5] It is the format Fig. showing the command D WORD suitable for using it in the desirable operation gestalt of this invention.
[Drawing 6] It is the explanatory view showing the channel discernment table suitable for using it in the desirable operation gestalt of this invention.
[Drawing 7] It is the explanatory view showing an example of the program address table suitable for using it in the desirable operation gestalt of this invention.
[Drawing 8] It is the flow chart which shows the setup and configuration procedure suitable for using it in the desirable operation gestalt of this invention.
[Drawing 9] It is the flow chart which shows the procedure of the data unit suitable for using it in the desirable operation gestalt of this invention.
[Description of Notations]
10 Code Processing System
12 Key Management Code Engine (KMCE)
13 Plaintext Interface Processor (PTIP)
14 Programmable Code Engine (PCE)
16 Code Engine Which Can be Constituted (CCE)
17 Programmable Code Processor (PCP)
18 Shared Memory
19 Code Control Section
20 Test Interface
21 Programmable Interface
22 ModN Solution Extractor (NSE)
24 Code Processor RAM
25 Internal Memory
26 Failsafe Reduced Instruction-set Computer (FS-RISC)

[Translation done.]